

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

**NÁVRH PI(D) REGULÁTORU PRO LINEÁRNÍ SYSTÉMY
NA ZÁKLADĚ AUTOMATICKÉHO SESTROJENÍ
OBLASTI STABILITY**

Bc. Marek Pakosta

Diplomová práce

2021

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2020/2021

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Marek Pakosta**
Osobní číslo: **I19303**
Studijní program: **N0714A150005 Automatické řízení**
Studijní obor: **Automatické řízení**
Téma práce: **Návrh PI(D) regulátoru pro lineární systémy na základě automatického sestrojení oblasti stability**
Zadávací katedra: **Katedra řízení procesů**

Zásady pro vypracování

Cílem diplomové práce je vytvoření algoritmů v jazyce MATLAB pro automatické sestrojení oblasti stability parametrů regulátoru PI a případně i PID na základě sestrojení hraniční křivky v rovině kp - ki a oddělení stabilních a nestabilních podoblastí. Získaná oblast stability bude graficky znázorněna pro zvolené typy lineárních systémů. Uvnitř oblasti stability bude dále vybrán jeden bod reprezentující optimální nastavení na základě vhodného kritéria. Chování regulačního obvodu bude ověřeno simulací.

Rozsah pracovní zprávy: **min. 60 stran**
Rozsah grafických prací:
Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

TAN, N., KAYA, I., YEROGLU, C., ATHERTON, D. P. *Computation of stabilizing PI and PID controllers using the stability boundary locus*. Energy Conversion and Management 47 (2006), 3045-3058.
KUBÍK, S., KOTEK, Z., STREJC, V., ŠTECHA, J. *Teorie automatického řízení I*. Praha: SNTL, 1982.

Vedoucí diplomové práce: **doc. Ing. Jan Cvejn, Ph.D.**
Katedra řízení procesů

Datum zadání diplomové práce: **6. listopadu 2020**

Termín odevzdání diplomové práce: **21. května 2021**

L.S.

Ing. Zdeněk Němec, Ph.D.
děkan

Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 16. listopadu 2020

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 1. 5. 2021

Bc. Marek Pakosta

PODĚKOVÁNÍ

Rád bych poděkoval svému vedoucímu, kterým byl doc. Ing. Jan Cvejn, Ph.D., za nabídnutí zajímavého tématu, rady a návrhy k řešení vzniklých komplikací, připomínky a zvýšení míry poznání, také bych rád poděkoval svým rodičům za podporu během studia.

V Pardubicích dne 1. 5. 2021

Bc. Marek Pakosta

ANOTACE

Tato práce se zabývá konkrétní implementací jedné z moderních metod umožňujících sestavení oblasti stabilních parametrů PI a PID regulátoru pro daný lineární systém. Implementovaná metoda spočívá ve výpočtu průběhu tzv. hraniční křivky, která svým průběhem rozdělí rovinu (k_P, k_I) nastavitelných parametrů regulátoru na stabilní a nestabilní podoblasti. Vymezené podoblasti jsou odděleny, individuálně reprezentovány a graficky znázorněny. Na základě testování stability vnitřního bodu jsou oblasti rozděleny na stabilní a nestabilní. Získaná oblast stability je dále využita pro optimální návrh PI regulátoru prostřednictvím minimalizace integrálního kritéria. Výsledky návrhu PI regulátoru jsou ukázány na zvolené sadě testovacích systémů. Provedeno je také rozšíření pro PID regulátor, kde nastavitelné parametry regulátoru leží v prostoru parametrů (k_P, k_I, k_D) . Úskalí, která byla v rámci implementace potřeba řešit, a jejich řešení jsou v průběhu práce systematicky diskutována.

KLÍČOVÁ SLOVA

PID regulátor, oblast stability, kvadratická regulační plocha, syntéza ve frekvenční oblasti, D-dekompozice.

TITLE

DESIGN OF THE PI(D) CONTROLLER FOR LINEAR SYSTEMS BASED ON AUTOMATIC CONSTRUCTION OF THE STABILITY DOMAIN.

ANNOTATION

This thesis presents an implementation of a modern approach for calculation of stability domain for PI and PID controllers. The implemented method is based on calculating and plotting stability boundary locus which divides the PI controller parameters plain (k_P, k_I) into stable and unstable regions. These regions are separated, represented individually and plotted graphically. The obtained stability domain is then used for the design of optimal PI controller using minimalization of integral criteria. Obtained results are then shown for a selected set of testing dynamical systems. Implemented method is then extended for the PID controller where the tunable parameters of the controller lie in the space of

(k_P, k_I, k_D) parameters. Complications regarding the implementation are systematically discussed throughout the thesis.

KEYWORDS

PID Controller, Stability Boundary Locus, Integral Criteria Design, Frequency Domain Synthesis, D-decomposition.

OBSAH

	SEZNAM ZKRATEK	10
	SEZNAM SYMBOLŮ PROMĚNNÝCH VELIČIN A FUNKCÍ	11
	SEZNAM OBRÁZKŮ	16
	ÚVOD	21
1	AUTOMATICKÉ ŘÍZENÍ	23
1.1	DYNAMICKÝ SYSTÉM	24
1.2	OBRAZOVÝ PŘENOS	27
1.3	DYNAMICKÉ CHARAKTERISTIKY V ČASOVÉ OBLASTI	28
1.4	REGULAČNÍ OBVOD	31
1.5	ANALÝZA VE FREKVENČNÍ OBLASTI	34
1.5.1	Amplitudo-fázová frekvenční charakteristika	35
1.5.2	Amplitudová a fázová frekvenční charakteristika	38
1.6	STABILITA	39
1.6.1	Algebraická kritéria stability	42
1.6.2	Frekvenční kritéria stability	45
1.6.3	Rezerva ve stabilitě	46
1.7	PID REGULÁTOR	49
1.7.1	Požadavky na regulaci	53
1.8	SYNTÉZA REGULAČNÍHO OBVODU S PID REGULÁTOREM	53
1.8.1	Syntéza v časové oblasti	54
1.8.2	Návrh dle umístění pólů přenosu	60
1.8.3	Syntéza ve frekvenční oblasti	62
2	NÁVRH PI(D) REGULÁTORU NA ZÁKLADĚ SESTROJENÍ OBLASTI STABILITY	66
2.1	METODA HRANIČNÍ KŘIVKY	67
2.2	REALIZACE METODY HRANIČNÍ KŘIVKY	73
2.2.1	Výpočet hraniční křivky	73
2.2.2	Ukončení výpočtu průběhu hraniční křivky	78
2.2.3	Úprava přenosu uvažovaného systému	84
2.2.4	Získání významných průsečíků průběhu hraniční křivky	87
2.2.5	Využití teorie grafů v souvislosti s metodou hraniční křivky	94

2.2.6	Reprezentace jednotlivých oblastí	104
2.2.7	Zajištění topologického uspořádání oblastí	108
2.2.8	Vyplnění oblasti	121
2.2.9	Nalezení stabilních oblastí	123
2.3	VYUŽITÍ OBLASTI STABILITY PRO OPTIMÁLNÍ NÁVRH PI REGU- LÁTORU	125
2.4	ROZŠÍŘENÍ O PARAMETR DERIVAČNÍ SLOŽKY PID REGULÁTORU ZÁVĚR	131 139
	POUŽITÁ LITERATURA	142
	SEZNAM PŘÍLOH	144

SEZNAM ZKRATEK

AFFCh	amplitudo-fázová frekvenční charakteristika
AFCh	amplitudová frekvenční charakteristika
DS	dynamický systém
FFCh	fázová frekvenční charakteristika
HK	hraniční křivka
IAE	absolutní regulační plocha
IE	lineární regulační plocha
ICH	impulsní charakteristika
ISE	kvadratická regulační plocha
ITAE	absolutní regulační plocha násobená časem
KU	koncový uzel (uzly)
ORO	otevřený regulační obvod
PCH	přechodová charakteristika
PID	proporcionálně integračně derivační (regulátor)
RP	regulační pochod
SU	standardní uzel (uzly)
URO	uzavřený regulační obvod

SEZNAM SYMBOLŮ PROMĚNNÝCH VELIČIN A FUNKCÍ

a	vážený koeficient PID regulátoru se dvěma stupni volnosti
a_i	koeficienty levé strany diferenciální rovnice
A_w	modul (amplituda) frekvenčního přenosu řízení
b	vážený koeficient PID regulátoru se dvěma stupni volnosti
b_i	koeficienty pravé strany diferenciální rovnice
$C(s)$	obrazový přenos ovládacího resp. regulujícího subsystému (regulátoru)
d	Eukleidovská vzdálenost
$D(s)$	jmenovatel obrazového přenosu systému
$d(t)$	poruchová veličina působící na výstupu soustavy
d_w	požadovaná hodnota euklidovské vzdálenosti
d_{\max}	maximální hodnota euklidovské vzdálenosti
e	hrana mezi dvěma vrcholy grafu, index označující sudé koeficienty frekvenčního přenosu systému
$ E(i\omega) $	modul spektra regulační odchylky
$E(s)$	Laplaceův obraz regulační odchylky
$e(t)$	regulační odchylka
\mathfrak{F}	operátor přímé Fourierovy transformace
$F(\omega)$	frekvenční přenos obecného systému F
\mathfrak{F}^{-1}	operátor inverzní Fourierovy transformace
$G(i\omega)$	frekvenční přenos ovládaného resp. regulovaného subsystému (soustavy)
$ G(i\omega) $	modul obrazového přenosu systému (amplituda), dB
$G(s)$	obrazový přenos ovládaného resp. regulovaného subsystému (soustavy)
$g(t)$	impulzní funkce
G_o	obrazový přenos ORO
H	Hurwitzova matice
h	hrana grafu
$H(s)$	Laplaceův obraz jednotkového skoku
$h(t)$	přechodová funkce
H_i	množina graf neorientovaného grafu

\Im	imaginární část
i	komplexní proměnná, pořadí kořene polynomu obrazového přenosu, řídicí proměnná
J	integrální kritérium (obecné)
j	komplexní, řídicí proměnná
J_{IAE}	integrální kritérium absolutní regulační plochy
J_{IE}	integrální kritérium lineární regulační plochy
J_{ISE}	integrální kritérium kvadratické regulační plochy
J_{ITAE}	integrální kritérium absolutní regulační plochy násobené časem
k	zesílení systému
k_P	váha proporcionální složky PID regulátoru, struktura s interakcí časových konstant, s
$k_P(\omega)$	funkce popisující průběh HK v ose k_I
$k_P(\omega)$	funkce popisující průběh HK v ose k_P
\mathcal{L}	operátor přímé L-transformace
$l(\omega)$	parametrická funkce popisující průběh HK
\mathcal{L}^{-1}	operátor inverzní L-transformace
m	řád pravé strany diferenciální rovnice, řídicí proměnná
m_A	amplitudová bezpečnost
M_S	citlivostní funkce
N	koefficient filtru derivační složky PID regulátoru
n	řád levé strany diferenciální rovnice, řídicí proměnná
$N(s)$	čitatel obrazového přenosu systému
O	oblast vymezená průběhem hraniční křivky
o	index označující liché koeficienty frekvenčního přenosu systému
o_r	relativní řád systému
P	reálná složka frekvenčního přenosu, Parsevalův integrál
p	znaménko poměru absolutních členů přenosu
$p(h)$	incidence hrany grafu
Q	imaginární složka frekvenčního přenosu
q	řád integračního resp. derivačního systému

$Q(\omega)$	substituce pro vyjádření parametrického vztahu popisujícího průběh hraniční křivky
\Re	reálná část
$R(\omega)$	substituce pro vyjádření parametrického vztahu popisujícího průběh hraniční křivky
R_i	úhlové oblasti neorientovaného grafu
r_{-1}	váha integrační složky PID regulátoru, struktura bez interakce časových konstant
r_0	váha proporcionální složky PID regulátoru, struktura bez interakce časových konstant
r_1	váha derivační složky PID regulátoru, struktura bez interakce časových konstant
s	komplexní proměnná
$S(\omega)$	substituce pro vyjádření parametrického vztahu popisujícího průběh hraniční křivky
s_0	výška ohraničujícího obdélníku („slabá“ mez)
s_1	výška ohraničujícího obdélníku („silná“ mez)
s_n	požadovaný počet bodů popisujících průběh hraniční křivky
t	čas, s
t_0	doba odezvy, s
T_d	dopravní zpoždění systému, s
T_n	doba náběhu, s
T_u	doba průtahu, s
t_{0m}	modifikovaná doba odezvy, s
T_{Dmax}	maximální hodnota derivační časové konstanty pro kterou ještě existuje rozeznatelná oblast vymezená průběhem HK, s
T_{Dmin}	minimální hodnota derivační časové konstanty pro kterou ještě existuje rozeznatelná oblast vymezená průběhem HK, s
T_D	derivační časová konstanta PID regulátoru, struktura s interakcí časových konstant, s
T_I	integrační časová konstanta PID regulátoru, struktura s interakcí časových konstant, s

t_{\max}	doba dosažení y_{\max} , s
T_{\min}	nejmenší časová konstanta obrazového přenosu systému
t_{reg}	doba regulace, s
$U(\omega)$	substituce pro vyjádření parametrického vztahu popisujícího průběh hraniční křivky
$U(s)$	Laplaceův obraz vstupní resp. akční veličiny
$u(t)$	vstupní resp. akční veličina
V	množina vrcholů
v	vrchol grafu
$v(t)$	poruchová veličina působící na vstupu soustavy
$w(t)$	žádaná resp. řídicí veličina
$X(\omega)$	substituce pro vyjádření parametrického vztahu popisujícího průběh hraniční křivky
$Y(\omega)$	substituce pro vyjádření parametrického vztahu popisujícího průběh hraniční křivky
$Y(s)$	Laplaceův obraz výstupní resp. regulované veličiny
$y(t)$	výstupní resp. regulovaná veličina
y_{\max}	nejvyšší hodnota regulované veličiny při překmitu
α_S	vzdálenost přímky absolutního útlumu od imaginární osy
γ	fázová bezpečnost, úhel přímky relativního útlumu, °
$\Delta(s)$	charakteristická rovnice resp. charakteristický polynom
$\eta(t)$	jednotkový skok
κ	váhový koeficient derivace regulované veličiny pro upravené kvadratické kritérium
μ	relativní překmit
ξ	součinitel relativního tlumení
τ	časová konstanta systému, čas, s
φ	fáze, úhel, °
ϕ	počet kvadrantů Gaussovy komplexní roviny, které AFFCh svým průběhem protne
ω	úhlová rychlost, $\text{rad}\cdot\text{s}^{-1}$
ω_c	kritická úhlová frekvence, $\text{rad}\cdot\text{s}^{-1}$

δ	pásma pro posouzení doby regulace
$\delta(t)$	Diracův impulz
ω_{max}	koncová hodnota úhlové frekvence pro vyjádření průběhu hraniční křivky, rad.s ⁻¹
ω_{min}	počáteční hodnota úhlové frekvence pro vyjádření průběhu hraniční křivky, rad.s ⁻¹
ω_n	vlastní úhlová frekvence systému, rad.s ⁻¹
ω_R	rezonanční převýšení, rad.s ⁻¹

SEZNAM OBRÁZKŮ

1.1	Příklad statické charakteristiky nelineárního systému	25
1.2	Dynamické charakteristiky vybraných typů dynamických systémů (17), (18), (19) z hlediska PCH a ICH, kde PDS, IDS, DDS je proporcionální, integrační resp. derivační dynamický systém, PČ, IČ, DČ je proporcionální, integrační resp. derivační člen a PNF je proporcionální neminimálně fázový systém	30
1.3	Struktura otevřeného regulačního obvodu, ovládací subsystém je v práci značen $C(s)$, ovládaný $G(s)$	32
1.4	Struktura uzavřeného regulačního obvodu, regulující subsystém je v práci značen $C(s)$, regulovaný $G(s)$	32
1.5	Příklady AFFCh vybraných typů dynamických systémů (41), kde A označuje systém 1. řádu, B – neminimálně fázový systém, C1 – systém 3. řádu a D – systém 3. řádu s dopravním zpožděním	38
1.6	Příklady AFCh sady proporcionálních dynamických systémů (41), A označuje systém 1. řádu, B – neminimálně fázový systém, C1, C2 – systém 3. řádu o různých koeficientech q a D – systém 3. řádu s dopravním zpožděním	39
1.7	Příklady FFCh sady proporcionálních dynamických systémů (41), A označuje systém 1. řádu, B – neminimálně fázový systém, C1, C2 – systém 3. řádu o různých koeficientech q a D – systém 3. řádu s dopravním zpožděním	39
1.8	A) stabilní stav, B) stav na mezi stability, C) nestabilní stav (Richard C. Dorf, 2016)	40
1.9	Experimentální letoun X-29 (Gazdoš F., 2013)	41
1.10	A) stabilní kmitavé chování, B) kmitavá mez stability, C) nestabilní kmitavé chování	42
1.11	Posouzení stability URO pomocí Nyquistova kritéria	46
1.12	Určení bezpečnosti v amplitudě a ve fázi na základě průběhu AFFCh	47
1.13	Modul citlivostní funkce pro (55)	48
1.14	Určení bezpečnosti v amplitudě a ve fázi na základě AFCh a FFCh	49
1.15	A – paralelní struktura, B – paralelní struktura s interakcí časových konstant, C – sériová struktura PID regulátoru	51
1.16	Ukazatele kvality RP v časové oblasti, (Richard C. Dorf, 2016)	56

1.17 Průběh regulační odchylky jednotlivých integrálních kritérií (68), (69), (70), (71) a příslušná regulační plocha	58
1.18 Příímka absolutního útlumu (Balátě, 2003)	61
1.19 Oblast rozložení kořenů charakterizována k požadovanému průběhu přechodového děje (Balátě, 2003)	62
1.20 Průběhy modulu přenosu řízení URO, 1 – nevhodný průběh, 2 – požadovaný průběh (Balátě, 2003)	63
1.21 Požadovaný průběh modulu přenosu řízení URO (Balátě, 2003)	63
2.1 Příklad oblasti stability v rovině (k_P, k_I)	66
2.2 Příklad průběhu hraniční křivky	68
2.3 Oblasti vymezené průběhem HK	68
2.4 Ukázka demonstrující segment průběhu HK jako sadu nastavení, které vedou na kmitavou mez stability. A – výběr tří sad nastavitelných parametrů regulátoru, B – AFFCh otevřeného regulačního obvodu odpovídající vybraným 3 sadám nastavení, C – regulační pochody získané pomocí vybraných 3 sad nastavení	69
2.5 Ilustrace zachycující obecné kroky získání stabilní oblasti a vnitřního bodu reprezentujícího optimální nastavení	72
2.6 Vypočítaný průběh HK (A) odpovídající průběhu frekvence ω (B)	74
2.7 Stanovení následujícího bodu průběhu HK na základě splnění podmínky euklidovské vzdálenosti (110)	75
2.8 Rychlostní profil $\omega(i)$, zaznamenány jsou ty hodnoty, které odpovídají podmínce (110), ty jsou označeny jako $\omega(k)$	76
2.9 A – průběh bodů HK se zajištěním jejich přibližných rozestupů, B – znázorňuje chybu jako rozdíl mezi referenční d_w a skutečnou euklidovskou vzdáleností d , C – znázorňuje odpovídající průběh argumentu ω	77
2.10 AFFCh, hledání průsečíků na základě FFCh	79
2.11 Ukázka nevhodného ukončení vypočítanou (nalezenou) $\omega_c = 14, 24$ (značena fialovou hvězdou) – průběh HK by dále pokračoval ve vymezení oblastí	79
2.12 Oblast O_{AC}, O_{BC} a O_{BD} , které se nachází mezi dvojicemi průsečíků A, B, C, D s osou $k_I = 0$	81
2.13 Ohraničující obdélník pro parametr s_0 (podmínka (116), resp. (117)) a parametr s_1 (podmínka (118))	81

2.14	Koncový bod HK vyhovující podmínce (116), resp. (117) ($s_0 = 10^3$ není v obrázku vyznačena)	82
2.15	Koncový bod HK vyhovující podmínce (118), pro $s_1 = 10^3$	82
2.16	A – ukončovací frekvenci nelze na úrovni výpočtu nalézt nebo je příliš vysoká, B – stabilní oblast vymezení až průběh hraniční křivky za ω_c (fialová hvězda)	83
2.17	Ilustrace průběhu HK systémů s pouze jedním pólem	83
2.18	FFCh původního (nerozšířeného) a FFCh rozšířeného systému	85
2.19	Průběh HK původního (122) a rozšířeného (123) systému, A – nekorektní ukončení průběhu původního systému a srovnání míry přesnosti s průběhem HK rozšířeného systému, B – korektní ukončení průběhu HK rozšířeného systému	86
2.20	Pořadí bodů reprezentujících významné průsečíky	87
2.21	A – zachycuje průběh HK a jednotlivé vymezené oblasti, B – ilustruje rozsahy indexů průsečíků reprezentujících jednotlivé oblasti v poli popisujícím průběh HK	88
2.22	Znázornění průsečíku 2 lineárních segmentů	89
2.23	Nesprávné (A) a správné řazení průsečíků (B)	91
2.24	Průsečíky popsané prostřednictvím nalezených indexů a zajištění sekvenčního seřazení	93
2.25	Multigraf	95
2.26	Prostý graf	95
2.27	Obyčejný graf	95
2.28	Hledané oblasti	96
2.29	A – multigraf, B – odpovídající virtualizovaný obyčejný graf	97
2.30	Hledané oblasti	98
2.31	Regulární oblast mezi více než dvěma KU (D)	98
2.32	A – planární graf mezi KU, B – odpovídající virtualizovaný planární graf mezi KU	99
2.33	Ukázka planárního grafu	101
2.34	A – multigraf, B – virtualizovaný multigraf (z hlediska hrany mezi KU – vrcholy 2 a 3), C – virtualizovaný obyčejný graf, D – virtualizovaný planární graf (z hlediska hrany mezi KU – vrcholy 2 a 3)	103
2.35	Oblast P vedoucí na kombinatorickou úlohu	104

2.36	Oblasti odpovídající různým kombinacím hran (142)	106
2.37	A – špatná orientace a pořadí segmentů křivky, B – správná orientace a pořadí segmentů křivky	106
2.38	A – průběh HK vedoucí na chybnou reprezentaci vymezených podoblastí, B – chybná reprezentace hranice oblasti O_1 a jí odpovídající chybné vyplnění (D), C – správná reprezentace hranice oblasti O_1 a jí odpovídající správné vyplnění (E)	109
2.39	Průběh HK definující vnější oblast typu smyčka	110
2.40	Průběh HK zahrnující nový typ oblasti z hlediska topologie	111
2.41	Nesprávná (A, B, C) a správná (A_T, B_T, C_T) reprezentace jednotlivých oblastí, které vymeží průběh HK na obr. 2.40	112
2.42	Určení nových bodů na základě vektorového součtu	113
2.43	Princip určení vnitřních bodů z množiny nových bodů na vstupu funkce	114
2.44	Výběr m vnitřních bodů z množiny vypočítaných nových bodů	114
2.45	Testovaný bod chybně označený jako vnitřní bod	115
2.46	Průběh HK hypotetického systému a znázornění jednotlivých vymezených oblastí	116
2.47	Správné zobrazení jednotlivých oblastí hypotetického systému č. 2	118
2.48	A a související A_1, A_2 zachycují špatnou reprezentaci z hlediska dat, B a související B_1, B_2 zachycují správnou reprezentaci z hlediska dat.	119
2.49	Princip konstrukce y -ových hodnot mřížky	121
2.50	Princip hledání bodů mřížky, které jsou uvnitř oblasti	122
2.51	Ukázka vyplněné oblasti	122
2.52	Nestabilní body na hranici stabilní oblasti	124
2.53	Nesprávné (A, vlivem rozlišení) a správné (B) odstranění vnitřních bodů v určité vzdálenosti podél hranice oblasti	124
2.54	Chybné označení oblasti vlivem nepřesnosti	125
2.55	Regulační pochody a oblast stability testovacího systému (149) – A	128
2.56	Regulační pochody a oblast stability testovacího systému (149) – B	128
2.57	Regulační pochody a oblast stability testovacího systému (149) – C	129
2.58	Regulační pochody a oblast stability testovacího systému (149) – D	129
2.59	Regulační pochody a oblast stability testovacího systému (149) – E	130
2.60	Regulační pochody a oblast stability testovacího systému (149) – F	130

2.61	Vliv derivační složky na deformaci průběhu HK (Tan et al., 2006)	133
2.62	Příklad deformace průběhů HK pro různé hodnoty parametru T_D , jehož důsledkem je kolaps oblastí stability	134
2.63	Příklad deformace oblasti stability pro různé hodnoty parametru k_D	134
2.64	Příklad deformace oblasti stability pro různé hodnoty parametru T_D	135
2.65	Příklad deformace oblasti stability pro různé hodnoty parametru T_D	135
2.66	Příklad deformace oblasti stability pro různé hodnoty parametru T_D	135
2.67	Regulační pochody systému (149) – C odpovídající optimálním nastavitelným parametrům PID regulátoru v prostoru parametrů (k_P, k_I, k_D)	136
2.68	Regulační pochody nestabilního systému (149) – F odpovídající optimálním nastavitelným parametrům PID regulátoru v prostoru parametrů (k_P, k_I, k_D) . . .	137

ÚVOD

PID regulátory jsou v průmyslových aplikacích zastoupeny až v 95 % případů zejména pro svoji robustnost, jednoduchost a možnost experimentálního nastavení s tím, že většinu z nich tvoří struktura PI, jelikož derivační složka nebývá využívána příliš často. Z tohoto důvodu má návrh PID regulátoru významnou roli (Tan, 2005). Syntézu regulačního obvodu s PI(D) regulátorem lze provádět v časové, komplexní anebo frekvenční oblasti.

V současné době existuje velké množství metod, které se návrhem PI(D) regulátoru zabývají. Neexistuje však jedna univerzální metoda a jednotlivé metody se liší svojí použitelností. Tyto metody lze rozdělit na empirické a analytické. Empirické metody vycházejí z uzavřeného regulačního obvodu a pracují s reálnou soustavu a reálným regulátorem. Praktickou výhodou těchto metod je, že nevyžadují znalost modelu – často bývají aplikovány na již existující regulační obvody, které je potřeba doladit nebo seřídit (Vítečková M., 2011). Analytické metody předpokládají znalost modelu soustavy (typicky obrazového přenosu) a představují postupy, pomocí kterých lze dosáhnout kvalitnějších regulačních dějů. Analytické metody návrhu lze dále rozdělit na syntézu v časové, komplexní a frekvenční oblasti.

Naprostá většina těchto metod se zabývá získáním jedné sady nastavitelných parametrů regulátoru – jednoho bodu v prostoru parametrů, který z pohledu dané metody představuje optimální nastavení.

Existuje však samostatná oblast výzkumu, která se zabývá nalezením celé množiny přípustných nastavení z hlediska stability. Tyto metody, které se v posledních desetiletích objevují v odborné literatuře, umožňují nalézt celou množinu stabilizujících nastavitelných parametrů regulátoru – tzv. oblast stability.

Zajištění stability představuje základní požadavek na regulační pochod, který je potřeba zajistit ještě před adresováním požadavků ostatních. Z toho důvodu je získání oblasti stability velice výhodné a lze ji dále využít k návrhu PI(D) regulátoru. Návrh PI(D) regulátoru na základě sestavení oblasti stability spočívá v hledání bodu v rovině parametrů, který z hlediska daného kritéria představuje optimální nastavení. K tomuto účelu lze využít i efektivní optimalizační algoritmy. Oblast stability lze však využít také v rámci robustního řízení.

Další významnou výhodou představuje znalost oblasti stability u nestabilních systémů. Mnoho procesů v industriální sféře totiž vykazuje nestabilní chování (reaktory, destilační

sloupce, zážehové systémy, ...). Některé systémy, např. ve vojenské či letecké technice, jsou dokonce navrženy tak, aby vykazovaly nestabilní chování za účelem zvýšení jejich manévrovatelnosti a rychlosti řízení. Experimentální hledání stabilizujících parametrů nestabilního systému může být velice náročné a pracné, důsledkem čehož je pro procesní inženýry mnohem obtížnější navrhnout vhodné řízení nestabilních systémů, jelikož se musí dbát na omezení plynoucí z jejich fyzikální podstaty. Z těchto důvodů může být znalost oblasti stabilizujících parametrů PI(D) regulátoru velice výhodná a praktická (Gazdoš F., 2013).

Jedna z výše uvedených moderních metod, které umožňují získání oblasti stability, je v této práci použita. Princip metody spočívá ve výpočtu a vykreslení průběhu tzv. hraniční křivky v rovině (k_P, k_I) . Jedná se o posloupnost hodnot k_P, k_I závislých na frekvenci ω . Hraniční křivka svým průběhem vymezení oblasti v rovině (k_P, k_I) v případě regulátoru PI a v prostoru parametrů (k_P, k_I, k_D) v případě regulátoru PID. Přičemž oblasti takto vymezené mohou být buď nestabilní nebo stabilní.

Tato práce se zabývá algoritmizací a konkrétní implementací uvedené metody. Autoři publikované metody se totiž ve svém článku (Tan et al., 2006) již konkrétní implementací nezabývali. Zejména se jedná o detaily numerického výpočtu průběhu hraniční křivky, nalezení a oddělení jednotlivých podoblastí, jejich individuální reprezentaci a grafické znázornění. Dále se článek (Tan et al., 2006) nezabývá hledáním optimálních nastavitelných parametrů PI(D) regulátoru v případě, že je daná oblast stabilní.

Kromě výše uvedené metody hraniční křivky, využívající tzv. D-dekompozice, existuje několik dalších teoretických přístupů, které umožňují nalézt stabilizující oblast. Patří mezi ně např. přístup spočívající v hledání tzv. singulárních frekvencí a zobecněné Hermitově-Biehlerově větě (Bajcinca, 2006), (Silva et al., 2001).

1 AUTOMATICKÉ ŘÍZENÍ

Systémem rozumíme obecně soubor prvků, mezi nimiž existují vzájemné vztahy a jako celek má vztahy ke svému okolí. Každý systém lze charakterizovat 2 základními vlastnostmi (Balátě, 2003):

- chováním systému – charakterizuje vnější vztahy systému k okolí, jedná se o závislost mezi podněty působící na vstup systému a příslušnými odezvami na jeho výstupu;
- strukturou systému – charakterizuje vnitřní funkční vztahy uvnitř systému, strukturou je myšlen jednak způsob uspořádání (organizace) vzájemných vazeb mezi prvky systému a jednak chování těchto prvků.

U systémů hraje roli rozlišovací úroveň – každý definovaný systém může být prvkem systému vyššího řádu (nadsystému), podobně mohou být prvky systému současně systémem nižšího řádu (subsystému) (Balátě, 2003).

Systém lze reprezentovat jako množinu

$$S = \{A_i, W_i, K, Q\}, \quad (1)$$

kde A je množina prvků a_i ($i = 1, 2, \dots, p$) systému S , tj. $A = \{a_1, a_2, \dots, a_p\}$,

W – množina vstupních veličin w_j , $W = \{w_1, w_2, \dots, w_m\}$,

Q – množina výstupních veličin q_h , $Q = \{q_1, q_2, \dots, q_n\}$,

K – množina vazebních závislostí k_{sr} výstupních a vstupních veličin prvků systému, kde index s značí výstupní prvek a index r vstupní prvek vazební závislosti (Balátě, 2003).

Jestliže jsou množiny W, Q prázdné, jde o absolutně uzavřený systém, v ostatních případech jde o relativně uzavřený systém, což vyjadřuje, že systém jako celek komunikuje se svým okolím prostřednictvím vstupů a výstupů. Množina A je nazývána jako univerzum systému. Soubor množin W, K, Q je nazýván charakteristikou systému (Balátě, 2003).

V teorii řízení lze systémy třídit podle (Balátě, 2003):

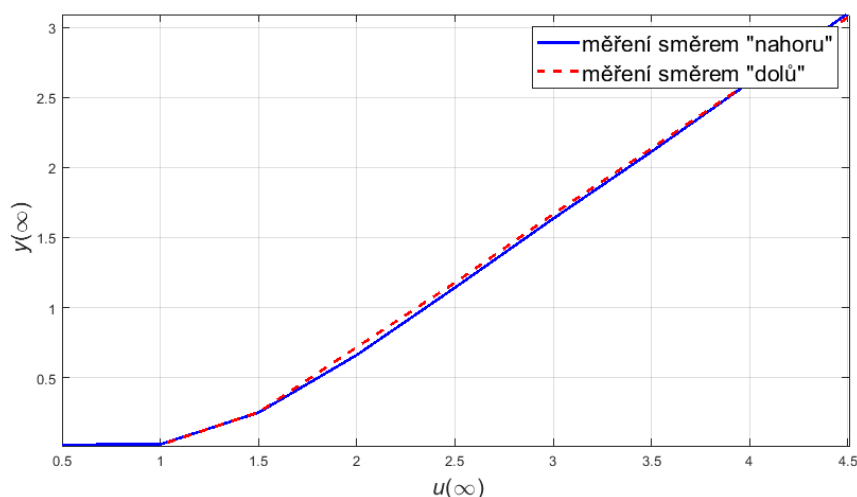
- Způsobu zpracování informace – systémy mechanické, pneumatické, hydraulické, elektro-mechanické, elektronické, optické, biologické, apod.
- Způsobu chování – rozlišujeme systémy:
 - deterministické – chování systému je jednoznačně určeno jeho stavem a vstupním signálem,

- stochastické – stav a vstup takového systému nejsou (vlivem jejich stochastického charakteru) známé. Jsou však známé jejich statistické charakteristiky,
- lingvistické (angl. fuzzy) – chování systému je popsáno pomocí fuzzy množin (vícehodnotovou logikou).
- Časového charakteru signálů – spojitě (analogové) a nespojitě (diskrétní).
- Závislosti změny chování na čase – závislé (angl. time variant), nezávislé (angl. time invariant).
- Typu matematického popisu – lineární a nelineární. Lineární systém lze popsat pomocí soustavy lineárních diferenciálních rovnic, pro tyto systémy platí princip superpozice.
- Počtu vstupů, funkce, počtu regulovaných veličin, sémantické hodnoty prvků, apod.

1.1 DYNAMICKÝ SYSTÉM

Pojem *dynamický systém* zdůrazňuje, že daný systém má nějaké dynamické vlastnosti, které určují vývoj jeho chování v čase, ve skutečnosti však mají tyto vlastnosti všechny reálné systémy. Toto tvrzení lze zdůvodnit akumulací hmoty anebo energie. Dynamické chování je vyjádřeno závislostí mezi vstupním a výstupním signálem systému (průběh vstupního signálu se často nazývá vzruchem a průběh výstupního odezvou) a grafické znázornění se nazývá *dynamická charakteristika* (Balátě, 2003).

Statické vlastnosti lze chápat jako limitní případ vlastností dynamických (po odeznění přechodového děje). Statické systémy (takové, jejichž stav se časově nevyvíjí) v přírodě neexistují, zjednodušením za ně však lze považovat systémy s velice nízkým stupněm dynamiky, např. elektronický zesilovač. Grafickým znázorněním statických vlastností je *statická charakteristika*. Předpoklad konstrukce statické charakteristiky je, že daný systém má ustálený stav, což nestabilní systémy (patří sem i systémy s integračním charakterem) nesplňují (Balátě, 2003). Příklad statické charakteristiky nelineárního systému je na obr. 1.1, nelinearita v tomto případě není jen v zesílení, ale vystupuje zde také nelinearita typu hystereze.



Obr. 1.1 – Příklad statické charakteristiky nelineárního systému

V této práci jsou uvažovány lineární systémy s časově neměnnými parametry. U nelineárních diferenciálních rovnic totiž neplatí princip superpozice a řešení takových rovnic je velmi obtížné. Také se předpokládá deterministická vazba mezi vstupem, stavem a výstupem systému. Dynamické chování systému ve spojitě časové oblasti je popsáno prostřednictvím *diferenciální rovnice* v obecném tvaru

$$\begin{aligned} a_n y^n(t) + a_{n-1} y^{n-1}(t) + \dots + a_1 y^1(t) + a_0 y(t) = \\ = b_m u^m(t) + b_{m-1} u^{m-1}(t) + \dots + b_1 u^1(t) + b_0 u(t), \end{aligned} \quad (2)$$

kde a_i jsou konstantní koeficienty levé strany diferenciální rovnice,
 b_i – konstantní koeficienty pravé strany diferenciální rovnice,
 $u(t)$ – vstupní veličina,
 $y(t)$ – výstupní veličina (Balátě, 2003).

Z podmínky fyzikální realizovatelnosti (příčina – následek) musí platit

$$m \leq n, \quad (3)$$

tuto podmínku lze rozlišovat jako

$$\begin{aligned} m = n \dots \text{slabá,} \\ m < n \dots \text{silná,} \end{aligned} \quad (4)$$

kde řád diferenciální rovnice n určuje řád systému (Balátě, 2003).

S výhodou je v teorii řízení k řešení diferenciálních rovnic využíváno *Laplaceovy transformace* (L-transformace) (Dušek, 2019), (Honc, 2018), ta umožňuje podstatně jednodušší analytické řešení a je definována jako

$$\begin{aligned} Y(s) &= \mathfrak{L}\{y(t)\} = \int_0^{\infty} y(t)e^{-st} dt, \\ Y(t) &= \mathfrak{L}^{-1}\{Y(s)\} = \frac{1}{2\pi i} \int_{c-i\omega}^{c+i\omega} Y(s)e^{st} ds, \end{aligned} \quad (5)$$

kde s je komplexní proměnná,

t – reálná proměnná,

$y(t)$ – tzv. originál (funkce definovaná pro $t \in \langle 0, \infty \rangle$),

$Y(s)$ – tzv. obraz (funkce definovaná v oblasti komplexní proměnné s),

\mathfrak{L} – operátor přímé L-transformace,

\mathfrak{L}^{-1} – operátor inverzní L-transformace.

Mezi omezení L-transformace patří to, že transformovaná funkce musí být integrovatelná, tzn. musí být po částech spojitá a musí růst pomaleji než exponenciála.

Základní vlastnosti L-transformace jsou:

- Linearita

$$a_1 \mathfrak{L}\{y_1(t)\} \pm a_2 \mathfrak{L}\{y_2(t)\} = a_1 Y_1(s) \pm a_2 Y_2(s). \quad (6)$$

- Posunutí v originále (časové zpoždění) a v obraze

$$\mathfrak{L}\{y(t-a)\} = e^{-as}Y(s), \quad \mathfrak{L}\{e^{\pm at}\} = Y(s \pm a). \quad (7)$$

- Věta o počáteční a koncové hodnotě

$$\begin{aligned} \lim_{t \rightarrow 0^+} y(t) &= \lim_{s \rightarrow \infty} sY(s), \\ \lim_{t \rightarrow \infty} y(t) &= \lim_{s \rightarrow 0} sY(s). \end{aligned} \quad (8)$$

- Obraz integrace

$$\mathfrak{L}\left\{\int_0^t y(t) dt\right\} = \frac{1}{s}Y(s). \quad (9)$$

- Obraz derivace

$$\mathfrak{L}\left\{\frac{d}{dt}y(t)\right\} = sY(s) - y(t=0). \quad (10)$$

1.2 OBRAZOVÝ PŘENOS

Pomocí L-transformace lze vyjádřit tzv. *obrazový přenos* systému $G(s)$, ten je definován jako poměr Laplaceova obrazu výstupní a vstupní veličin při nulových počátečních podmínkách. Diferenciální rovnici (2) lze pomocí L-transformace transformovat na tvar

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + \dots + b_1 s + b_0}{a_n s^n + \dots + a_1 s + a_0}. \quad (11)$$

Přičemž polynomy v čitateli a jmenovateli přenosu lze rozložit na součin kořenových součinitelů

$$\begin{aligned} a_n s^n + \dots + a_1 s + a_0 &= a_n (s - s_1) \dots (s - s_i) \dots (s - s_n), \\ b_m s^m + \dots + b_1 s + b_0 &= b_m (s - n_1) \dots (s - n_j) \dots (s - n_m), \end{aligned} \quad (12)$$

kde s_i , ($i = 1, 2, \dots, n$) jsou póly – kořeny jmenovatele, které mohou být buď reálné, nebo komplexně sdružené,

s_j , ($j = 1, 2, \dots, m$) jsou nuly – kořeny čitatele, mohou být taktéž reálné, či komplexně sdružené (Balátě, 2003).

Přenos systému (11) lze pomocí pólů a nul vyjádřit ve tvaru

$$G(s) = \frac{b_m (s - n_1) \dots (s - n_j) \dots (s - n_m)}{a_n (s - s_1) \dots (s - s_j) \dots (s - s_n)}. \quad (13)$$

Rozdíl stupně polynomu ve jmenovateli a čitateli přenosu se nazývá *relativní řád systému*

$$o_r = n - m. \quad (14)$$

Záporně vzatým převráceným hodnotám reálných pólů a nul říkáme *časové konstanty* systému. Jsou-li póly i nuly reálné, lze přenos systému vyjádřit pomocí časových konstant ve tvaru

$$G(s) = \frac{b_0 (1 + s\tau_1)(1 + s\tau_2) \dots (1 + s\tau_m)}{a_0 (1 + s\tau_1)(1 + s\tau_2) \dots (1 + s\tau_n)}, \quad (15)$$

a poměr

$$k = \frac{b_0}{a_0} \quad (16)$$

udává tzv. *zesílení systému* (Balátě, 2003).

Lze rozlišovat následující typy dynamických systémů:

- Proporcionální dynamický systém – existuje nenulový ustálený stav. Obecný proporcionální dynamický systém se setrvačností n -tého řádu má tvar

$$G(s) = \frac{b_m s^m + \dots + b_1 s + b_0}{a_n s^n + \dots + a_1 s + a_0}. \quad (17)$$

- Integrační dynamický systém – také se označují jako systémy s astatismem. Pro tyto systémy neexistuje ustálený stav, za ustálený stav lze považovat průběh jiné stavové veličiny (např. rychlost, zrychlení, apod.). Obecný tvar integračního dynamického systému q -tého řádu se setrvačností n -tého řádu je

$$G(s) = \frac{b_m s^m + \dots + b_1 s + b_0}{s^q (a_n s^n + \dots + a_1 s + a_0)}. \quad (18)$$

- Derivační dynamický systém – ustálený stav existuje, ale je nulový. Obecný tvar derivačního dynamického systému q -tého řádu se setrvačností n -tého řádu je

$$G(s) = \frac{s^q (b_m s^m + \dots + b_1 s + b_0)}{a_n s^n + \dots + a_1 s + a_0}. \quad (19)$$

Odezvy vybraných typů dynamických systémů jsou na obr. 1.2 (Balátě, 2003).

1.3 DYNAMICKÉ CHARAKTERISTIKY V ČASOVÉ OBLASTI

Chování systému lze charakterizovat jeho odezvou na standardizovaný signál přivedený na jeho vstup, mezi standardizované signály v časové oblasti patří např. Heavisideův (resp. jednotkový) skok nebo Diracův (resp. jednotkový) pulz (Balátě, 2003).

Posuzujeme-li odezvu systému na jednotkový skok, potom hovoříme o přechodové funkci $h(t)$, jejíž znázorněním je *přechodová charakteristika* (PCH), jednotkový skok se značí $\eta(t)$ (někdy také $1(t)$) a je definován jako

$$u(t) = \eta(t) \begin{cases} 1 & \text{pro } t \geq 0, \\ 0 & \text{pro } t < 0, \end{cases} \quad (20)$$

přičemž Laplaceův obraz jednotkového skoku je definován jako

$$\mathfrak{L}\{\eta(t)\} = \mathfrak{L}\{1(t)\} = \frac{1}{s}, \quad (21)$$

potom obraz přechodové funkce je

$$\mathfrak{L}\{h(t)\} = H(s) = G(s)U(s) = \frac{G(s)}{s}. \quad (22)$$

Hodnotu přechodové funkce lze vyjádřit v čase $t = \infty$ jako (Balátě, 2003)

$$h(\infty) = \lim_{s \rightarrow 0} sH(s) = \lim_{s \rightarrow 0} s \frac{G(s)}{s} = \lim_{s \rightarrow 0} G(s). \quad (23)$$

Přechodová funkce systému 1. řádu nebo pseudo 1. řádu (relativní řád takového systému je $o_r = 1$) má v počátku nenulovou derivaci a příslušná tečna přímo vymezí časovou konstantu systému, na rozdíl od systému vyššího řádu, který má v počátku nulovou derivaci s tím, že pro určitou segmentaci přechodové funkce $h(t)$ je potřeba vést sečnu v inflexním bodě ta vymezí dobu průtahu T_u a dobu náběhu T_n . Dobu průtahu lze použít pro následnou aproximaci systému vyššího řádu systémem 1. řádu s dopravním zpožděním T_d .

Systém s neminimální fází (nestabilní nuly v čitateli) o kladném statickém zesílení má, v závislosti na počtu nul v pravé polorovině komplexní roviny, typicky zápornou derivaci v počátku. PCH vybraných typů dynamických systémů (včetně neminimálně fázových) jsou uvedeny na obr. 1.2.

Impulzní charakteristika (ICH) je znázornění impulzní funkce $g(t)$, která je odezvou na jednotkový (Diracův) impulz. Diracův impulz je definován jako

$$\int_{-\infty}^{\infty} \delta(t) dt = 1, \quad \delta(t) = 0 \text{ pro } t \neq 0. \quad (24)$$

Laplaceův obraz Diracova impulzu je

$$\mathfrak{L}\{\delta(t)\} = 1. \quad (25)$$

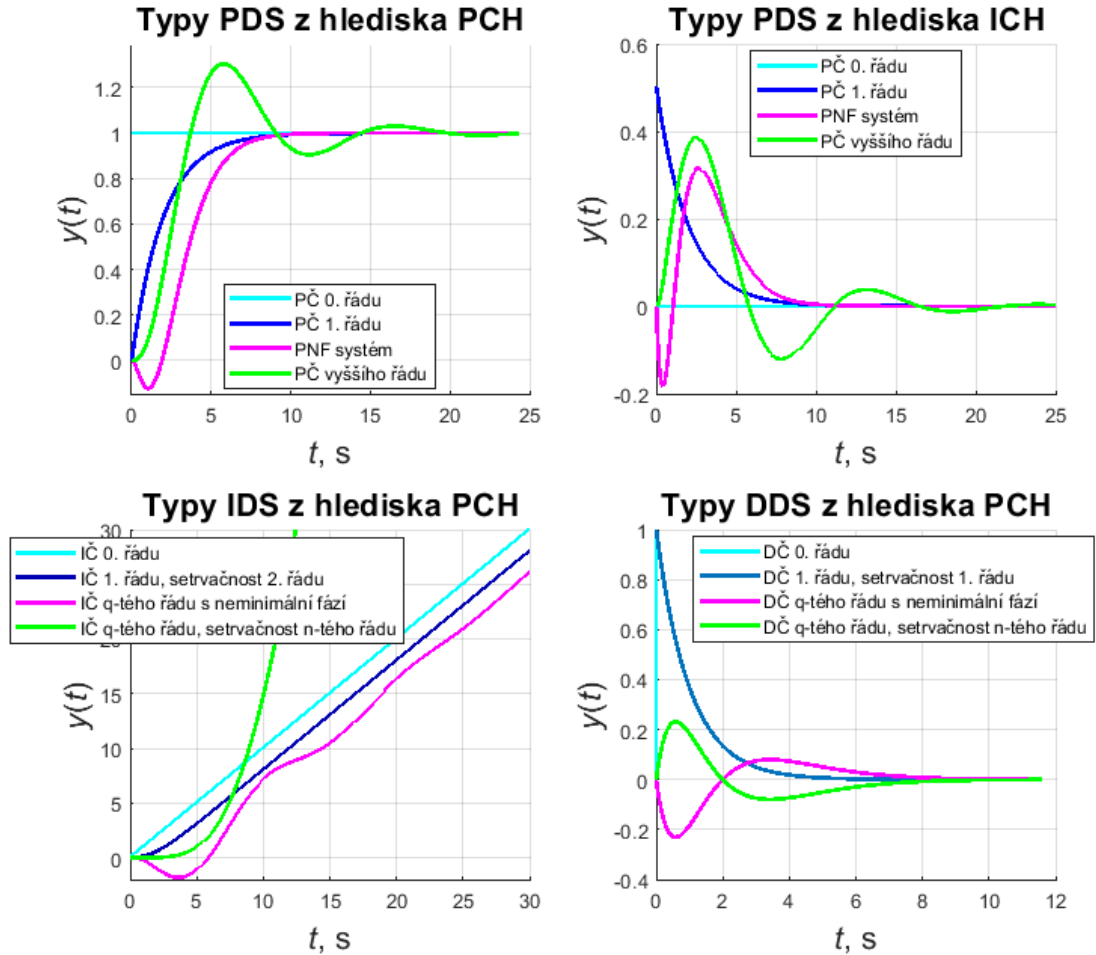
Obraz impulzové funkce je

$$\mathfrak{L}\{g(t)\} = G(s)\mathfrak{L}\{\delta(t)\} = G(s) \quad (26)$$

a rovná se přenosu systému (Balátě, 2003).

Má-li systém $o_r = 0$, potom jednotkový impulz projde systémem až na výstup. Je-li $o_r = 1$, potom má impulzní funkce v čase $t = 0$ nespojitost, tj. skok z hodnoty $g(t = 0) = 0$ na $g(t = 0) = \frac{b_{n-1}}{a_n}$, je-li $o_r \geq 2$, potom impulzní funkce začíná v 0. ICH vybraných typů

dynamických systémů (včetně neminimálně fázových) jsou uvedeny na obr. 1.2 (Balátě, 2003).



Obr. 1.2 – Dynamické charakteristiky vybraných typů dynamických systémů (17), (18), (19) z hlediska PCH a ICH, kde PDS, IDS, DDS je proporcionální, integrační resp. derivační dynamický systém, PČ, IČ, DČ je proporcionální, integrační resp. derivační člen a PNF je proporcionální neminimálně fázový systém

Mezi PCH a ICH existuje vzájemný vztah

$$h(t) = \int_0^t g(\tau) d\tau, \quad g(t) = \frac{dh(t)}{dt}, \quad (27)$$

umožňující přepočít.

Podobně, je-li systém popsán impulzní funkcí a chceme-li získat odezvu na obecný vstupní signál $u(t)$, lze pro tento záměr využít konvoluci – pro obraz výstupního signálu

platí $Y(s) = G(s)U(s)$ a tedy pro originál odezvy bude, v případě nulových počátečních podmínek $y(t)$, platit (Balátě, 2003)

$$y(t) = \int_0^t g(t - \tau)u(\tau)d\tau, \quad (28)$$

resp.

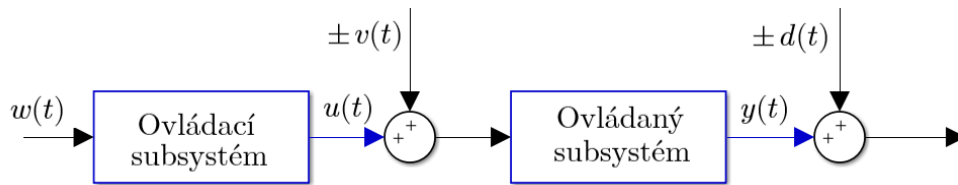
$$y(t) = \int_0^t g(\tau)u(t - \tau)d\tau. \quad (29)$$

1.4 REGULAČNÍ OBVOD

Řízení (Honc, 2018), (Franklin Gene F., 2015) zahrnující osobu, která ovládá zařízení, se nazývá manuální řízení, naopak řízení, které probíhá bez účasti člověka, se nazývá automatické řízení. *Regulátor* je zařízení, které má zajistit konstantní hodnotu $y(t)$ i přes působící poruchy, nebo sledování $w(t)$. Regulační obvod vzniká připojením regulátoru k regulované soustavě, výstupní veličinou regulačního obvodu je regulovaná veličina $y(t)$, vstupní veličinou je řídicí veličina a poruchová veličina $v(t)$ působící na vstupu soustavy, resp. $d(t)$ působící uvnitř nebo na výstupu soustavy, tyto poruchy mohou být obecně kladné nebo záporné. Jednotlivé typy signálů jsou znázorněny na obr. 1.3 resp. 1.4.

Proces řízení spočívá v cíleném působení na řízený systém s cílem zajištění co nejvyšší míry shody mezi výstupní veličinou $y(t)$ a vstupní veličinou $u(t)$. Řídicí systémy lze rozlišit z hlediska informace, která je použita k výpočtu akčního zásahu, na otevřené a uzavřené (zpětnovazební) regulační obvody.

Otevřený regulační obvod (ORO viz obr. 1.3) nepracuje s měřenou výstupní veličinou $y(t)$. Výhodou ovládání je jeho jednoduchost, kvalita závisí na míře porozumění danému systému a vlivu poruch. Nelze však tímto způsobem zajistit stabilitu. Systém ovládání sestává z ovládacího a ovládaného subsystému, ovládaná žádaná hodnota na vstupu je ovládacím subsystémem přepočítána a výstupní veličina přímo působí na ovládaný subsystém, kromě změny žádané hodnoty mají na výstup vliv také poruchy působící na vstupu, resp. uvnitř, a na výstupu ovládaného subsystému.



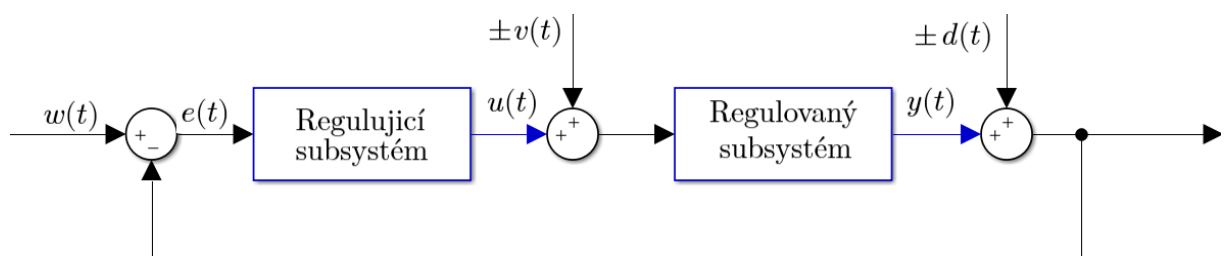
Obr. 1.3 – Struktura otevřeného regulačního obvodu, ovládací subsystém je v práci značen $C(s)$, ovládaný $G(s)$

Pokud je do regulátoru zpětnou vazbou přivedena informace o aktuální hodnotě výstupní veličiny a ten na základě této informace vypočítává akční zásah, tak se jedná o řízení v uzavřené smyčce – *uzavřený regulační obvod* (URO), nebo také zpětnovazební řízení – viz obr. 1.4.

Systém regulace sestává z regulujícího a regulovaného subsystému, vstupem regulátoru je v tomto případě rozdíl $w(t)$ a $y(t)$, který je přes zpětnou vazbu přiváděn do rozdílového členu. Opět je potřeba uvažovat poruchy působící na vstupu, resp. uvnitř, a na výstupu soustavy. Nesoulad mezi požadovanou a skutečnou hodnotou na výstupu představuje regulační odchylka

$$e(t) = w(t) - y(t). \quad (30)$$

Smyslem řízení je, aby akční veličina působila na regulovanou soustavu tak, aby $y \rightarrow w$, nebo ekvivalentně $e \rightarrow 0$.



Obr. 1.4 – Struktura uzavřeného regulačního obvodu, regulující subsystém je v práci značen $C(s)$, regulovaný $G(s)$

Efektivnímu vykompenzování poruch působících na vstupu a na výstupu soustavy nelze vyhovět současně. Porucha působící na výstupu soustavy má vyšší vliv na regulační pochod – má podobnou dynamiku jako změna $w(t)$. Regulační obvod nastavený pro rychlé vyrovnání poruch působících na vstupu soustavy často reaguje podstatně hůře

při skokových změnách žádané hodnoty (poruchy působící na výstupu soustavy) a naopak (Vítečková M., 2011).

Kromě zpětnovazebního řízení je potřeba rozlišit ještě *dopředné řízení*, to umožňuje použít měřenou informaci o poruchách ke korekci výpočtu akčního zásahu a tím kompenzovat jejich vliv, ještě než se projeví na regulované veličině.

Ovládaný nebo řízený subsystém může být popsán svým modelem. *Model* představuje umělý systém, který do určité míry napodobuje chování původního systému a musí vyjadřovat ty stránky daného jevu, které jsou z hlediska studia a zkoumání důležité. Proces tvorby modelu se nazývá *modelování* a dochází při něm k určité redukci vzhledem k původnímu systému, jedná se například o předpoklad soustředěných konstantních parametrů, lineárních závislostí, zanedbání malých vlivů, neurčitosti a šumu apod. Pro teorii automatického řízení má největší význam redukce systému na jeho matematický model. Proces hledání matematického popisu (modelu) se nazývá *identifikace* (Balátě, 2003).

Analytická identifikace označuje postup využívající matematicko-fyzikální analýzu. Matematický model se získá sestavením rovnic popisujících vztahy mezi vstupními, stavovými a výstupními signály prvků. Při sestavování rovnic se vychází ze zákona zachování hmoty, zákona zachování energie, z rovnic dynamické rovnováhy, vztahů kompatibility apod. Na počátku je potřeba zvolit vhodné veličiny a použít příslušné fyzikální zákony (síla-pohyb, proud-napětí, apod.). Tento způsob identifikace vyžaduje velké teoretické znalosti, uplatňování vhodných zjednodušujících předpokladů a aproximací. Výsledkem je popis v tzv. přirozeném stavu – je získán vnitřní model systému vyjadřující závislost mezi vstupními a stavovými veličinami a mezi stavovými a výstupními veličinami. Parametry tohoto modelu mají navíc fyzikální význam, to umožňuje kontrolu správnosti řešení, ekvivalence obou stran je nutnou (nikoli postačující) podmínkou správnosti řešení (Balátě, 2003).

Experimentální identifikaci je možné provádět nad experimentálně získanými daty. Mějme data o výstupu soustavy, které představují odezvu na příslušně vygenerovaný vstup – vstupní data. Poté lze hledat matematickou závislost, která popisuje vztah mezi vstupem a výstupem systému. Aby se výstup modelu co nejvíce podobal výstupu skutečného systému, je potřeba zavést nějaké kritérium – např. metodu nejmenších čtverců. Tímto způsobem je získán tzv. vstupně-výstupní popis systému – je popsána závislost mezi vstupní a výstupní proměnnou. Parametry často nemají fyzikální, ale pouze symbolický

význam, takto získaný model je typicky použitelný v menším rozsahu (např. v okolí pracovního bodu, v ustáleném stavu apod.) (Balátě, 2003).

S identifikací je také spojen pojem *verifikace*, při kterém dochází k testování míry shody chování systému s jeho modelem, je možné ji využít ke zpřesňování parametrů modelu (Balátě, 2003).

Jakmile je model systému k dispozici, je možné nahradit experimentování na reálném zařízení experimentováním na jeho modelu. Při experimentování s modelem je napodobována nějaká situace, nebo jsou vytvářeny podmínky, které mohou nastat u skutečného systému během normálních či havarijních podmínek. Tyto činnosti souhrnně označuje pojem *simulace* (Balátě, 2003).

1.5 ANALÝZA VE FREKVENČNÍ OBLASTI

V praxi se často posuzuje chování systémů v časové oblasti, ve které se využívají dynamické charakteristiky představující odezvy systémů na různě definované časově závislé vstupy. Analýza v časové oblasti má významnou roli a je intuitivní, nicméně stejně tak významná je i analýza v oblasti frekvenční (Golnaraghi M. F, 2017).

Analýza ve frekvenční oblasti se často využívá u komunikačních systémů, pro které má, oproti té časové, větší význam, jelikož většina signálů má buď sinusový průběh, nebo se z takových průběhů skládá. Mezi chováním systémů v časové a komplexní oblasti existuje analogie, na základě analýzy ve frekvenční oblasti je možné predikovat chování v oblasti časové. Tedy provedení analýzy ve frekvenční oblasti neimplikuje, že systém bude vystaven pouze vstupům s kmitavým charakterem, ale na základě reakce systému na kmitavý vstup je možné projektovat chování systému v čase. Frekvenční oblast je také vhodnější, chceme-li zkoumat citlivost systému na šum či změnu parametrů (Golnaraghi M. F, 2017).

Shrnutím lze konstatovat, že primární motivací analýzy ve frekvenční oblasti je dostupnost a vhodnost existujících analytických nástrojů, sekundárně tento přístup představuje jiný úhel pohledu na syntézu regulačních obvodů a jako takový často poskytuje cenné či dokonce stěžejní informace o vyšetřovaném systému (Golnaraghi M. F, 2017).

1.5.1 Amplitudo-fázová frekvenční charakteristika

V souvislosti s amplitudo-fázovou frekvenční charakteristikou (AFFCh) je nejprve potřeba uvést pojem *frekvenční přenos* (Cvejn, 2017). Frekvenční přenos je definován jako obrazový přenos v L-transformaci, kde se dosadí $s = i\omega$, tedy

$$F(\omega) = G(i\omega). \quad (31)$$

Pokud existují Fourierovy obrazy $Y(\omega)$ a $U(\omega)$, pak je frekvenční přenos

$$F(\omega) = \frac{Y(\omega)}{U(\omega)}. \quad (32)$$

Fourierova transformace je definována jako

$$Y(\omega) = \mathfrak{F}(y(t)) = \int_{-\infty}^{\infty} y(t)e^{-i\omega t} dt, \quad (33)$$

potom zpětná F-transformace

$$y(t) = \mathfrak{F}^{-1}(Y(\omega)) = \frac{1}{2\pi} \int_{-\infty}^{\infty} Y(\omega)e^{i\omega t} d\omega. \quad (34)$$

Kde $y(t)$ je originál,

$Y(\omega)$ – Fourierův obraz,

\mathfrak{F} – operátor přímé Fourierovy transformace,

\mathfrak{F}^{-1} – operátor zpětné (inverzní) Fourierovy transformace.

Příčemž Fourierův obraz má pro dané ω význam komplexní amplitudy harmonické složky signálu s frekvencí ω .

Na rozdíl od obrazového přenosu má frekvenční přenos i konkrétní fyzikální význam. Pokud do lineárního systému s časově neměnnými parametry vstupuje signál s danou frekvencí ω , tak na výstupu je signál o stejné frekvenci (v případě lineárních časově invariantních systémů), ale se změněnou amplitudou, anebo fázovým posuvem.

Dosazením konkrétní hodnoty ω v intervalu $\omega = \langle 0; \infty \rangle$ do frekvenčního přenosu $G(i\omega)$ je získán bod tzv. *amplitudo-fázové frekvenční charakteristiky* (AFFCh), tu lze znázornit jako křivku v komplexní rovině. Bod AFFCh lze také získat měřením – po přivedení harmonického signálu na vstup systému je po odeznění přechodových dějů změřena amplituda a fázový posuv výstupního signálu oproti signálu vstupnímu. Křivka AFFCh bývá v anglické literatuře označována jako tzv. křivka Nyquistova. V případě, že má

zkoumaný systém póly v pravé polorovině (je nestabilní, viz dále), však nedojde k odeznění přechodových složek a měření je tím nerealizovatelné. Potom je možné s frekvenční charakteristikou pracovat pouze matematickými prostředky (Cvejn, 2017).

Dále pro potřeby konstrukce frekvenční charakteristiky může být frekvenční přenos v kartézském či polárním tvaru. Kartézské vyjádření je uvažováno ve tvaru

$$G(i\omega) = P(\omega) + iQ(\omega) = \Re[G(i\omega)] + i\Im[G(i\omega)]. \quad (35)$$

V případě kartézského souřadného systému je na osu x vynášena hodnota \Re části a na osu y potom hodnota \Im části $|G(i\omega)|$. Z tohoto tvaru lze určit modul (vzdálenost bodů od počátku) jako

$$|G(i\omega)| = \sqrt{P(\omega)^2 + Q(\omega)^2} \quad (36)$$

a úhel

$$\varphi(\omega) = \arctan\left(\frac{Q(\omega)}{P(\omega)}\right), \quad (37)$$

který svírá úsečka z počátku do daného bodu v Gaussově rovině. Jsou-li hodnoty modulu a úhlu známy, je možné převést komplexní vyjádření do goniometrického tvaru – odpovídá transformaci na polární souřadnice. Lze vyjádřit

$$\Re[G(i\omega)] = |G(i\omega)| \cos(\varphi(\omega)); \Im[G(i\omega)] = |G(i\omega)| \sin(\varphi(\omega)). \quad (38)$$

Potom frekvenční přenos

$$\begin{aligned} G(i\omega) &= \Re[G(i\omega)] + i\Im[G(i\omega)] = \\ &= |G(i\omega)| \cos(\varphi(\omega)) + i|G(i\omega)| \sin(\varphi(\omega)) = \\ &= |G(i\omega)|(\cos(\varphi(\omega)) + i \sin(\varphi(\omega))). \end{aligned} \quad (39)$$

S využitím Eulerova vzorce $e^{ix} = \cos(x) + i \sin(x)$ je získán vztah

$$G(i\omega) = |G(i\omega)|e^{i(\varphi(\omega))}, \quad (40)$$

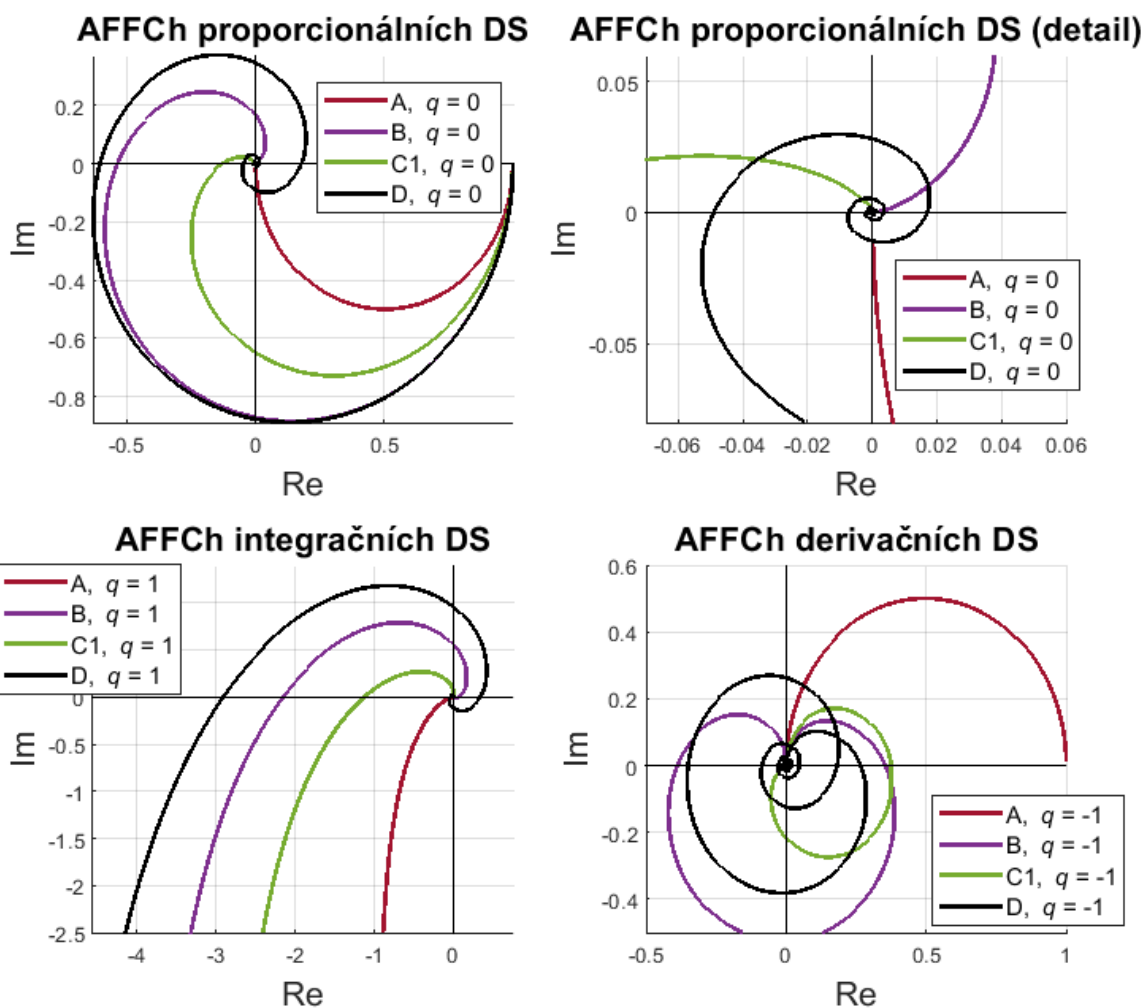
což je vyjádření frekvenčního přenosu v polárních souřadnicích – modul se označuje jako *amplituda* a úhel, který je funkcí frekvence, je označován jako *fázový posun*.

Na obr. 1.5, 1.6 a 1.7 je uvažována následující sada systémů

$$\begin{aligned}
 G_A(s) &= \frac{s^q}{s+1}, \\
 G_B(s) &= \frac{s^q(-1, 1s+1)}{s^3+3s^2+3s+1}, \\
 G_{C1}(s) &= \frac{s^q}{s^3+3s^2+3s+1}, \\
 G_{C2}(s) &= \frac{s^q}{0,2s^3+1,4s^2+2,2s+1}, \\
 G_D(s) &= \frac{s^q e^{-2,3s}}{s^3+3s^2+3s+1}.
 \end{aligned} \tag{41}$$

Hodnota fázového posuvu pro nekonečnou frekvenci se dá pomocí relativního řádu vypočítat jako

$$\phi(\omega) = -o_r \frac{\pi}{2}. \tag{42}$$



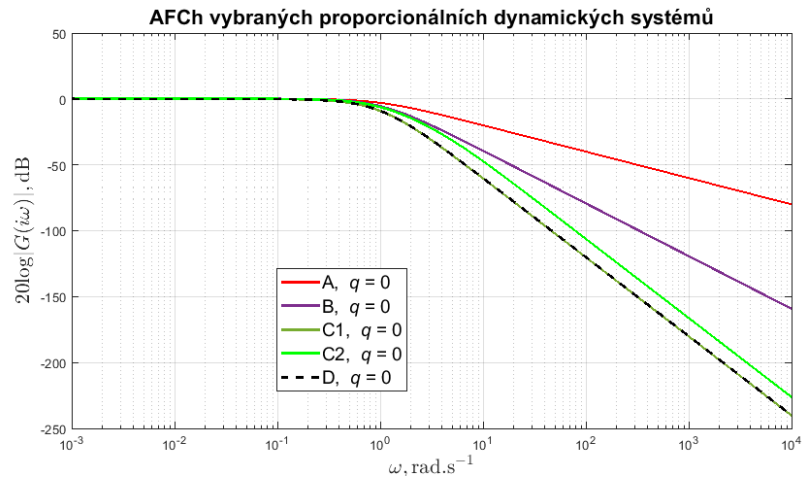
Obr. 1.5 – Příklady AFFCh vybraných typů dynamických systémů (41), kde A označuje systém 1. řádu, B – neminimálně fázový systém, C1 – systém 3. řádu a D – systém 3. řádu s dopravním zpožděním

1.5.2 Amplitudová a fázová frekvenční charakteristika

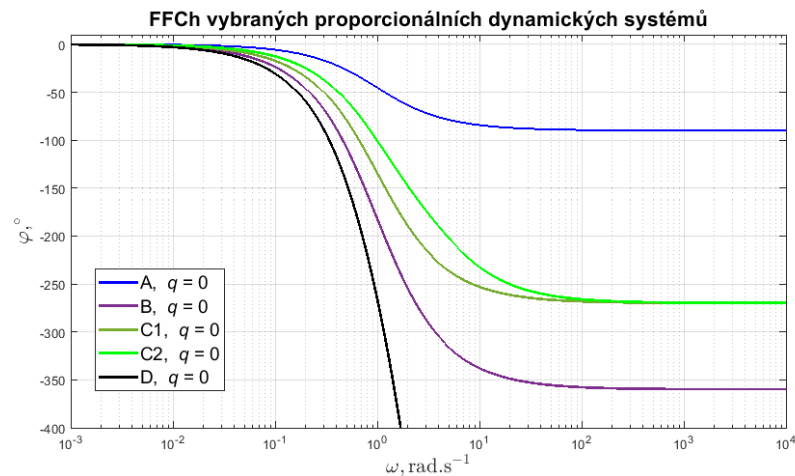
Ve frekvenční oblasti je v podstatě jenom jedna charakteristika, jmenovitě AFFCh. Ta, jak již bylo zmíněno výše, udává závislost zesílení systému a fázového posuvu výstupního signálu ku vstupnímu signálu vyjádřená v komplexní rovině.

AFFCh lze vyjádřit pomocí dvou samostatných charakteristik, a to pomocí amplitudové frekvenční charakteristiky (AFCh) – závislost zesílení systému na frekvenci vstupního signálu v ustáleném stavu (viz obr. 1.6) a fázové frekvenční charakteristiky (FFCh) – závislost fázového posuvu výstupního signálu ku vstupnímu signálu na frekvenci vstupního signálu v ustáleném stavu, viz obr. 1.7.

Výhoda AFCh a FFCh spočívá v tom, že je lze stanovit experimentálně (Dušek, 2019).



Obr. 1.6 – Příklady AFCh sady proporcionálních dynamických systémů (41), A označuje systém 1. řádu, B – neminimálně fázový systém, C1, C2 – systém 3. řádu o různých koeficientech q a D – systém 3. řádu s dopravním zpožděním

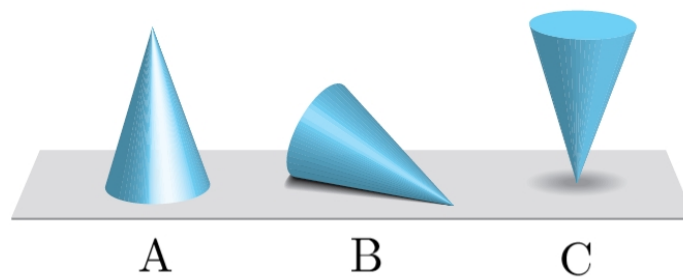


Obr. 1.7 – Příklady FFCh sady proporcionálních dynamických systémů (41), A označuje systém 1. řádu, B – neminimálně fázový systém, C1, C2 – systém 3. řádu o různých koeficientech q a D – systém 3. řádu s dopravním zpožděním

1.6 STABILITA

Pojem stabilita představuje v teorii řízení zásadní pojem. Je možné posuzovat stabilitu systému jako samostatné entity (subsystému, procesu) nebo systému ve smyslu regulačního obvodu. Mnoho procesů v industriální sféře vykazuje nestabilní chování (reaktory, destilační

sloupce, zážehové systémy, ...), stejně tak je spousta systémů v ekologické či sociální oblasti z podstaty nestabilní. Některé systémy, např. ve vojenské či letecké oblasti, jsou dokonce navrženy tak, aby vykazovaly nestabilní chování za účelem zvýšení manévrovatelnosti a rychlosti řízení. Důsledkem čehož je pro procesní inženýry mnohem obtížnější navrhnout vhodné řízení nestabilních systémů, jelikož se musí dbát na omezení plynoucí z jejich fyzikální podstaty. Z praktického pohledu nemá význam URO, který není stabilní. Důležitou roli hrají nástroje modelování a simulace, protože experimentování na reálném zařízení (nestabilním systému) je velice hazardní (Gazdoš F., 2013), (Richard C. Dorf, 2016).



Obr. 1.8 – A) stabilní stav, B) stav na mezi stability, C) nestabilní stav (Richard C. Dorf, 2016)

Pro posouzení stability (Cvejn, 2017) existuje více definic. Stabilitu neautonomních systémů lze definovat následovně. „*Systém je stabilní, pokud pro omezený vstup je omezený i výstup*“. Lineární systém je stabilní právě tehdy, když

$$\int_0^{\infty} |g(t)| dt < \infty, \quad (43)$$

a také tehdy když platí, že všechny póly přenosu (kořeny charakteristické rovnice – jmenovatele přenosu) leží v levé polorovině Gaussovy komplexní roviny. Toto tvrzení dává přímou metodu vyšetření stability na základě polohy pólů charakteristického mnohočlenu.

Je možné rozlišovat mezi absolutní a relativní stabilitou. Pojem *absolutní stabilita* vyjadřuje, jestli je systém stabilní či nikoliv. Dále je-li URO stabilní, potom je možné určit míru stability – *relativní stabilitu* – kupříkladu průkopníci návrhu letounů byli s tímto pojmem velice dobře obeznámeni – čím více vykazovalo chování letounu stabilnější chování, tím hůře se s ním manévrovalo. Prakticky umožnilo vyšetřování relativní stability dosáhnout vysoké míry manévrovatelnosti moderních letounů (akrobatický letoun vykazuje méně stabilní chování než komerční letadlo a díky tomu může manévrovat rychleji).

Relativní stabilitu je možné posoudit na základě umístění pólů URO. Je-li relativní stabilita posuzována v oblasti frekvenční, pak lze hovořit o *bezpečnosti* ve fázi a v amplitudě (Richard C. Dorf, 2016).



Obr. 1.9 – Experimentální letoun X-29 (Gazdoš F., 2013)

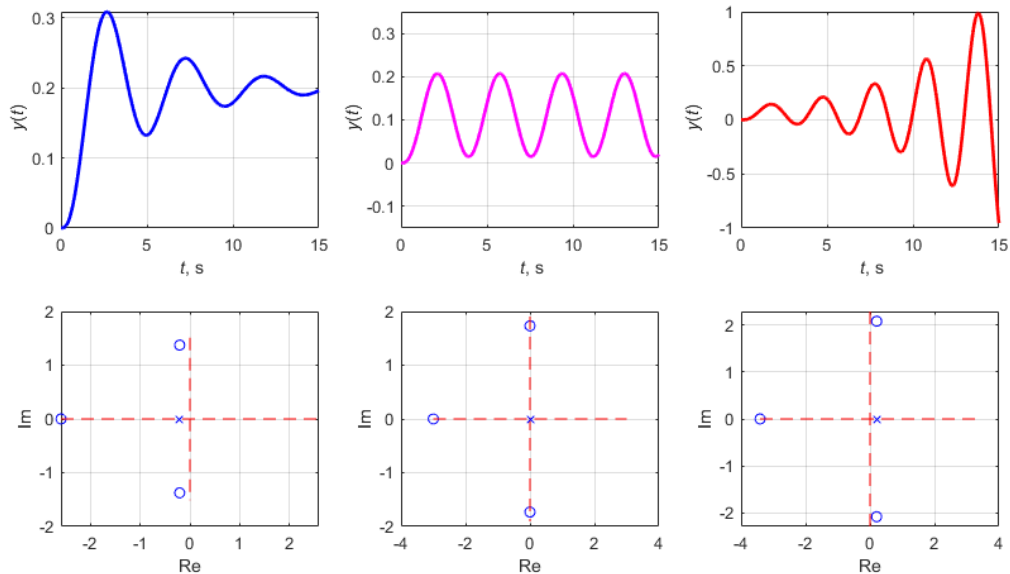
Teoreticky narůstá výstup nestabilního systému nade všechny meze, prakticky však vždy dojde k nějakému omezení (např. saturace nárůstu polohy vlivem kolize s překážkou nebo vyšší hladinou atmosféry u stíhacího letounu). Kromě stabilního a nestabilního systému je možné také definovat systém, který je tzv. na *mezi stability* (viz obr. 1.8 – B) – jedná se o stav systému způsobený určitým typem vstupního signálu. Takový systém je však už ve skutečnosti nestabilní, protože jeho póly leží na imaginární ose. Pro určitý tvar vstupního signálu však bude jeho výstup kmitat s konstantní amplitudou, ale amplituda kmitů se nebude zvětšovat ani zmenšovat.

Systém je možné destabilizovat volbou omezeného kmitavého vstupu takového, že jeho frekvence odpovídá velikosti komplexně sdružených kořenů na imaginární ose – stabilita takového systému je podmíněna a pouze některé omezené vstupy způsobí nestabilní chování. Uvažujme systém jehož charakteristická rovnice má tvar

$$\Delta(s) = (s + 10)(s^2 + 16) = 0, \quad (44)$$

kde řešení odpovídá reálný kořen $s_1 = -10$ a 2 kořeny komplexně sdružené $s_{2,3} = \pm 4i$, vybuzením tohoto systému sinusovou frekvencí $\omega = 4$ je tento systém destabilizován (Richard C. Dorf, 2016).

Stav, kdy je systém na mezi stability, prakticky nikdy nastat nemůže (v praxi nelze nastavit zcela přesné hodnoty parametrů zpětné vazby), ale u reálných systémů často nastává vlivem nelinearit



Obr. 1.10 – A) stabilní kmitavé chování, B) kmitavá mez stability, C) nestabilní kmitavé chování

Uzavřený regulační obvod je sám o sobě dynamický systém. Stabilita systému je základním požadavkem správného fungování a je potřeba ji při návrhu *zajistit ještě před* adresováním ostatních požadavků na regulační pochod. Z toho vyplývá, že nejprve jsou nalezeny stabilní parametry regulátoru, který zajistí stabilní chování URO, a až poté je možné, uvážlivou volbou stabilních parametrů, získat lepší jakost RP.

Právě hledání stabilních parametrů je obtížné, k tomuto účelu byly již historicky vyvinuty různé matematické postupy, ty se však liší z hlediska své efektivity, použitelnosti a složitosti, přičemž tato práce se zabývá konkrétní implementací jednoho z nich. Otázka získání stabilizujících parametrů regulátoru, které zajistí stabilitu URO, je však aktuální i v případě stabilních systémů – vhodnou volbou stabilních parametrů je totiž možné docílit dalších požadavků na regulační pochod (Richard C. Dorf, 2016).

1.6.1 Algebraická kritéria stability

Algebraická kritéria vyhodnocují polynom jmenovatele přenosu sledovaného systému a na základě manipulací s koeficienty polynomu dávají odpověď na otázku, zda existuje alespoň jeden kořen v pravé polorovině komplexní roviny (Dušek, 2019).

Algebraická kritéria nejsou použitelná pro systémy s dopravním zpožděním. Také neposkytují informaci o tom, do jaké míry je obvod tlumený. Nutnou podmínkou stability je, že *všechny koeficienty charakteristického polynomu musí existovat a mít stejné stejné*

znaménko. Pokud tato podmínka není splněna, pak systém není stabilní. Pokud tato podmínka splněna je, pak systém může (ale nemusí) být stabilní, z čehož vyplývá, že posoudit nestabilitu je z hlediska výpočetního času rychlejší (Balátě, 2003).

Nevýhodou přímé metody vyšetření stability založené na určení kořenů charakteristického mnohočlenu je nutnost řešení algebraické rovnice vyššího řádu, což je možné exaktně pouze pro řád $n \leq 4$ (pro vyšší řád nutno numericky) (Cvejn, 2017).

Existují však metody, které umožňují rozhodnout o stabilitě bez nutnosti řešit charakteristickou rovnici. Význam těchto metod v současné době poněkud klesá, protože pomocí moderních softwarových nástrojů není problém určit kořeny mnohočlenů libovolného řádu. Přesto jsou však důležité např. pro návrh regulátorů (Cvejn, 2017).

Hurwitzovo kritérium

Výchozí pro Hurwitzovo kritérium (Cvejn, 2017) stability je charakteristický polynom (resp. mnohočlen) ve tvaru

$$\Delta(s) = a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0. \quad (45)$$

kde a_i je koeficient charakteristického polynomu,

Z koeficientů charakteristického polynomu je sestavena tzv. Hurwitzova matice tak, že koeficienty charakteristické rovnice jsou rozděleny na sudé a liché a zapsány posunuté do řádků pod sebou podle schéma

$$\begin{bmatrix} a_{n-1} & a_{n-3} & a_{n-5} & 0 & 0 \\ a_n & a_{n-2} & a_{n-4} & 0 & 0 \\ 0 & a_{n-1} & a_{n-3} & 0 & 0 \\ 0 & a_n & a_{n-2} & 0 & 0 \\ 0 & 0 & 0 & a_2 & a_0 \end{bmatrix}. \quad (46)$$

Také se předpokládá, jak bylo uvedeno výše, že všechny koeficienty a_0, a_1, \dots, a_n jsou kladné (pokud jsou všechny záporné předpokládá se úprava vynásobením -1). Aby byl

system stabilní, musí být všechny subdeterminanty příslušné prvkům na hlavní diagonále Hurwitzovy matice sestavené z koeficientů charakteristické rovnice kladné, tedy

$$H_1 = a_{n-1}, H_2 = \begin{bmatrix} a_{n-1} & a_{n-3} \\ a_n & a_{n-2} \end{bmatrix}, H_3 = \begin{bmatrix} a_{n-1} & a_{n-3} & a_{n-5} \\ a_n & a_{n-2} & a_{n-4} \\ 0 & a_{n-1} & a_{n-3} \end{bmatrix}, \dots \quad (47)$$

až do řádu $n-1$. Pro soustavu 2. řádu ($n = 2$) zřejmě stačí, aby znaménka všech koeficientů byla kladná (Cvejn, 2017).

Kritérium není vhodné pro $n > 5$. V případě, že $a_0 = 0$, je jeden kořen v počátku souřadnic komplexní roviny $s_1 = 0$, potom se jedná o tzv. nekmitavou (aperiodickou) mez stability. Když $H_{n-1} = 0$, pak jsou dva kořeny ryze imaginární (leží na imaginární ose souměrně podle počátku souřadnic v komplexní rovině). V takovém případě se jedná o kmitavou mez stability.

Routhovo-Schurovo kritérium

Vychází z charakteristického polynomu (45) URO, předpokládá se, že a_0, a_1, \dots, a_n jsou kladné. Výhodou Routhova-Shurova kritéria stability je, že je dáno formou algoritmu a je vhodné i pro vyšetřování charakteristického polynomu vysokého stupně za předpokladu, že jeho koeficienty jsou zadány číselně (Balátě, 2003). Postup je následující (Balátě, 2003):

1. Koeficienty charakteristického polynomu (45) jsou napsány vedle sebe a seřazeny od nejvyšší či nejnižší mocniny.
2. Tato posloupnost koeficientů je rozdělena na sudou a lichou část.
3. Každý sudý koeficient je násoben podílem prvních dvou koeficientů a napsán pod předcházející řadu posunutou o jeden koeficient vlevo.
4. Tato nová řada koeficientů je odečtena od předcházející, čímž se posloupnost zkrátí o jeden člen (dojde k tzv. redukci).
5. Jsou-li všechny koeficienty v nové posloupnosti kladné, opakujeme postup.
6. Jakmile se při některé redukci objeví některý koeficient záporný, je výpočet ukončen se závěrem, že charakteristický polynom obsahuje nestabilní kořen – systém je nestabilní.
7. Dospějeme-li postupnou redukcí k řadě tří kladných koeficientů, je možné učinit závěr, že charakteristická rovnice (45) má všechny kořeny ve stabilní oblasti.

1.6.2 Frekvenční kritéria stability

Významnou výhodou kmitočtových kritérií je možnost posoudit stabilitu URO na základě frekvenční charakteristiky otevřené smyčky a jejich aplikovatelnost i na systémy s dopravním zpožděním. Mějme obrazový přenos ORO

$$G_o(s) = C(s)G(s), \quad (48)$$

kde $C(s)$ je obrazový přenos regulátoru,

$G(s)$ – obrazový přenos regulované soustavy, viz obr. 1.4.

Opět se předpokládá, že charakteristický polynom je vyššího nebo stejného řádu jako polynom v čitateli (slabá podmínka fyzikální realizovatelnosti) (Balátě, 2003).

Nyquistovo kritérium stability

Nyquistovo kritérium stability je velice praktickým kritériem pro posouzení stability, jelikož umožňuje posuzovat stabilitu URO na základě AFFCh otevřeného regulačního obvodu, ta může být k dispozici v podobě grafu či tabulky s experimentálně naměřenými daty – právě možnost jejího změření na reálném zařízení představuje další výhodu této metody (Balátě, 2003).

Předpokládá se, že přenos (48) neobsahuje žádné póly v pravé polorovině (tj. řízený systém je stabilní), kromě nanejvýš jednoho nulového pólu. Potom je pro stabilní ORO možné formulovat tzv. *zjednodušené Nyquistovo kritérium stability* jako „je-li otevřený regulační obvod stabilní, pak uzavřený regulační obvod bude stabilní tehdy a jen tehdy, jestliže AFFCh otevřeného regulačního obvodu neobklopuje kritický bod $[-1, i0]$ zleva“, viz obr. 1.11 (modře).

Příklad – mějme systém a regulátor popsany přenosem

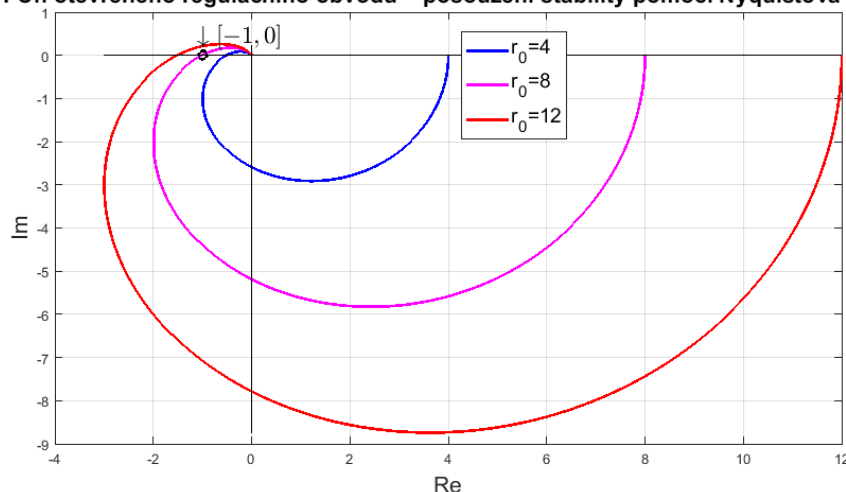
$$G(s) = \frac{1}{s^3 + 3s^2 + 3s + 1}, \quad C(s) = r_0. \quad (49)$$

Dále přenos ORO

$$G_o(s) = C(s)G(s) = \frac{r_0}{s^3 + 3s^2 + 3s + 1}. \quad (50)$$

AFFCh je znázorněna pro $r_0 = 4, 8, 12$ na obr. 1.11, kde je vidět, že volba $r_0 = 8$ způsobí přivedení URO na kmitavou mez stability (2 póly jsou komplexně sdružené).

AFFCh otevřeného regulačního obvodu – posouzení stability pomocí Nyquistova kritéria



Obr. 1.11 – Posouzení stability URO pomocí Nyquistova kritéria

Zobecněné Nyquistovo kritérium stability pro nestabilní ORO lze formulovat jako „Je-li ORO nestabilní a má-li jeho charakteristický polynom celkem p nestabilních kořenů, pak URO bude stabilní tehdy a jen tehdy, když Nyquistova křivka (křivka, která se skládá z AFFCh a jejího symetrického obrazu kolem reálné osy) otevřeného regulačního obvodu obklopí kritický bod $[-1, i0]$ $\frac{p}{2}$ -krát v kladném smyslu (tj. proti směru hodinových ručiček), kde jedno obklopení znamená změnu fázového posuvu o 2π “ (Balátě, 2003).

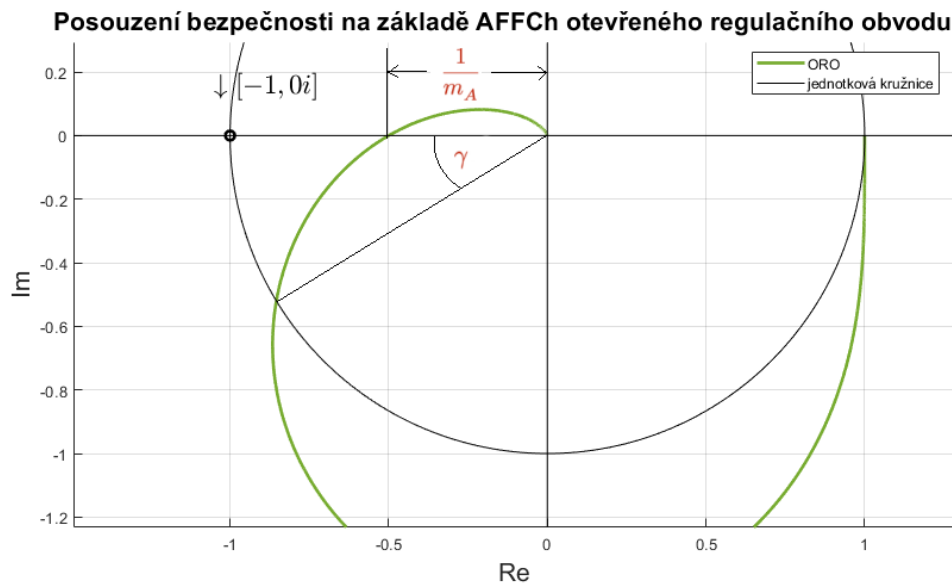
Z hlediska zobecněného Nyquistova kritéria, kořeny charakteristického polynomu (45) ORO ležící na imaginární ose nejsou považovány za nestabilní, tj. nepočítají se k těm, které leží v pravé polorovině komplexní roviny. Pokud např. charakteristický polynom ORO má q -násobný nulový kořen (tj. ORO obvod obsahuje q integračních členů), pak pro posouzení stability URO je potřeba AFFCh modifikovat ve smyslu spojení s kladnou reálnou poloosou obloukem o velikém (teoreticky nekonečném) poloměru v záporném smyslu (ve směru pohybu hodinových ručiček) s centrálním úhlem $q\frac{\pi}{2}$ (Balátě, 2003).

1.6.3 Rezerva ve stabilitě

Změnou parametrů ORO lze ovlivňovat míru stability URO. Z toho vyplývá, že je možné vyjádřit míru stability jako převrácenou hodnotu vzdálenosti průsečíku AFFCh (průsečíku s protifází) od počátku souřadného systému – *amplitudová bezpečnost* m_A . Dále je možné stanovit úhel mezi průsečíkem AFFCh s jednotkovou kružnicí se středem v počátku souřadného systému a zápornou reálnou poloosou. Získaný úhel poskytuje informaci, o jak velký úhel je možné zvětšit fázové zpoždění, aby při nezměněné amplitudě byl obvod

opět na hranici stability – *fázová bezpečnost* γ , jinými slovy – při jak velkém rozdílu fáze vzhledem ke kritické fázi -180° je vzdálenost od kritického bodu $[-1, 0i]$ nejmenší. Obvod jednotkové kružnice odpovídá $G(i\omega) = 1$, tzn. že v místě průsečíku je zesílení rovno jedné (Balátě, 2003).

Oba tyto parametry závisí na místě průsečíku AFFCh křivky se zápornou reálnou poloosou, AFFCh je však vyjádřením dvou veličin, z toho důvodu je potřeba posuzovat 2 parametry. Stabilitnímu URO bude odpovídat $m_A = (1, \infty)$ a $\gamma = (0, 2\pi)$ (Balátě, 2003).



Obr. 1.12 – Určení bezpečnosti v amplitudě a ve fázi na základě průběhu AFFCh

Dalším významným ukazatelem rezervy ve stabilitě je *maximum modulu citlivostní funkce* (Dušek, 2019). Citlivostní funkce odpovídá přenosu šumu, resp. přenosu regulační odchylky

$$G_{ew}(s) = \frac{1}{1 + C(s)G(s)}, \quad (51)$$

kde $C(s)$ je přenos regulátoru, $G(s)$ – přenos soustavy.

potom maximum modulu citlivostní funkce

$$M_S = \max_{\omega} |G_{ew}(i\omega)|. \quad (52)$$

Mezi modulem citlivostní funkce a amplitudovou a fázovou bezpečností platí

$$m_A > \frac{M_S}{M_S - 1}, \quad \gamma > 2 \arcsin\left(\frac{1}{2M_S}\right). \quad (53)$$

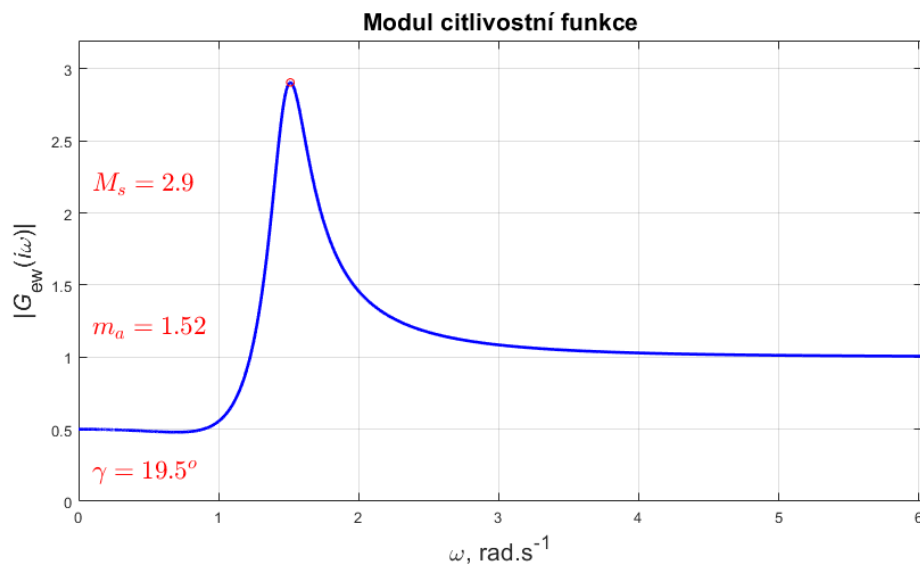
Příčemž doporučené hodnoty amplitudové a fázové bezpečnosti jsou

$$2 \leq m_A \leq 5, \quad \frac{\pi}{6} \leq \gamma \leq \frac{\pi}{3}. \quad (54)$$

Uvažujme přenos ORO

$$G(s) = \frac{3}{s^3 + 3s^2 + 3s + 1}. \quad (55)$$

Na základě maxima modulu citlivostní funkce M_S (52) bylo pomocí (53) resp. (54) stanoveno $m_A = 1,52$ a $\gamma = 19,5$, skutečná hodnota (dle obr. 1.12) však zhruba odpovídá $m_A \approx 2$, resp. $\gamma \approx 35^\circ$.



Obr. 1.13 – Modul citlivostní funkce pro (55)

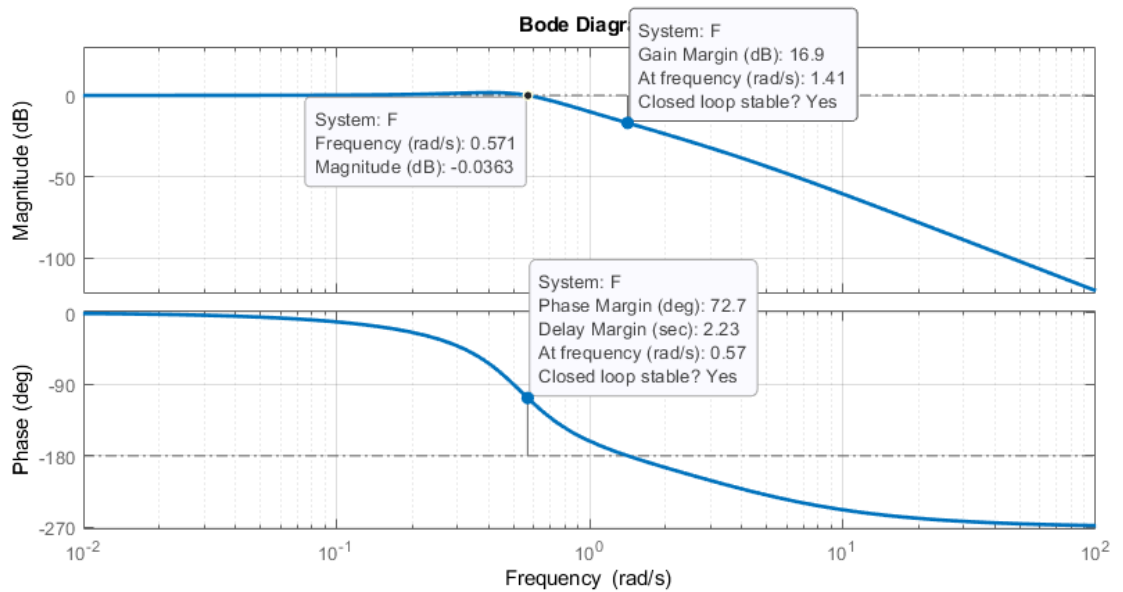
Bezpečnost v amplitudě m_A a ve fázi γ lze stanovovat také na základě AFCh resp. FFCh, označme ω_{co} jako frekvenci, při které

$$|G(i\omega_{co})| = 0 \text{ dB}, \quad (56)$$

potom bezpečnost ve fázi odpovídá

$$\gamma = \angle G(i\omega_{co}) + 180^\circ, \quad (57)$$

a bezpečnost v amplitudě m_A lze definovat jako zesílení potřebné k získání průsečíku AFCh s 0 dB v bodě, kde frekvence odpovídá poklesu FFCh o -180° .



Obr. 1.14 – Určení bezpečnosti v amplitudě a ve fázi na základě AFCh a FFCh

1.7 PID REGULÁTOR

V průmyslu má návrh PID regulátoru významnou roli, tyto regulátory jsou v průmyslových aplikacích často používány zejména pro svojí robustnost, jednoduchost a možnost experimentálního nastavení. Právě kvůli těmto vlastnostem jsou PID regulátory zastoupeny až z 95 % s tím, že většinu z nich tvoří struktura PI, jelikož D-složka není využívána velice často (Tan, 2005).

Proporcionálně integračně derivační regulátor (Honc, 2018) je struktura určená do URO, tedy do regulačního obvodu se zavedenou zápornou zpětnou vazbou. Výstup tohoto regulátoru nezávisí pouze na velikosti regulační odchylky, ale také na jejím integrálu a derivaci s tím, že každá z těchto složek reaguje na regulační odchylku s různou vahou – nastavitelné parametry regulátoru.

Lze rozlišit následující struktury PID regulátoru:

- Paralelní struktura bez interakce časových konstant (viz obr. 1.15 – A) – chování této struktury popisuje rovnice

$$u(t) = r_0 e(t) + r_{-1} \int_0^t e(\tau) d\tau + r_1 \frac{de(t)}{dt}, \quad (58)$$

kde r_0 je váha proporcionální složky regulátoru,

r_{-1} – váha integrační složky regulátoru,

r_1 – váha derivační složky regulátoru.

Aplikací L-transformace na (58) a následnou úpravou lze získat přenos regulátoru

$$C(s) = \frac{U(s)}{E(s)} = r_0 + \frac{r_{-1}}{s} + r_1 s. \quad (59)$$

- Paralelní struktura s interakcí časových konstant (viz obr. 1.15 – B) – tuto strukturu lze získat předřazením bloku zesílení. Chování popisuje rovnice

$$u(t) = k_P \left[e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right]. \quad (60)$$

kde k_P je váha proporcionální složky regulátoru,
 T_I – integrační časová konstanta regulátoru,
 T_D – derivační časová konstanta regulátoru.

Aplikací L-transformace na (60) a následnou úpravou lze získat

$$C(s) = \frac{U(s)}{E(s)} = k_P \left[1 + \frac{1}{T_I s} + T_D s \right]. \quad (61)$$

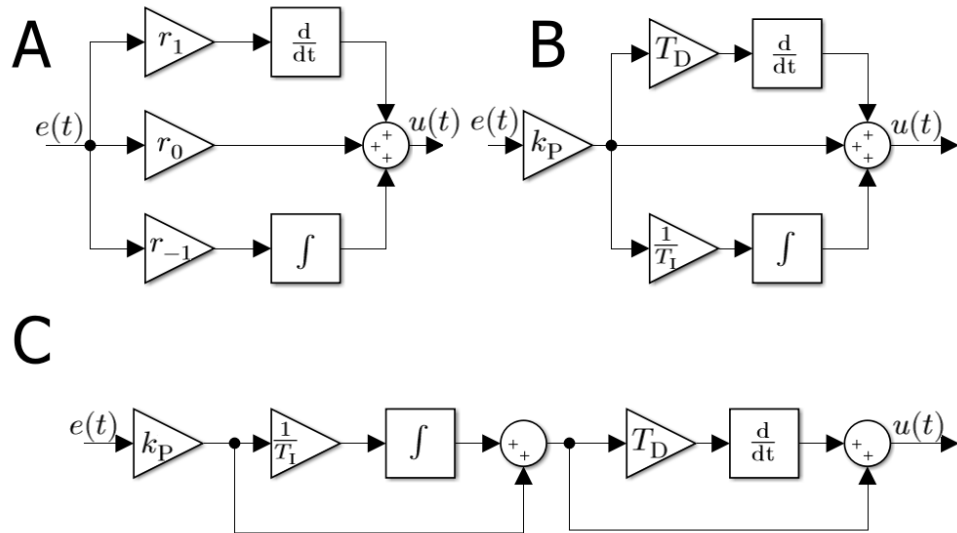
- Sériová struktura (viz obr. 1.15 – C) – Chování této struktury popisuje rovnice

$$u(t) = k_P \left[e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau \right] \left[1 + T_D \frac{de(t)}{dt} \right]. \quad (62)$$

Aplikací L-transformace na (62) a následnou úpravou lze získat

$$C(s) = \frac{U(s)}{E(s)} = k_P \left[1 + \frac{1}{T_I s} \right] [1 + T_D s]. \quad (63)$$

Nastavitelné parametry regulátoru je potřeba volit tak, aby zajistily požadovanou kvalitu regulačního pochodu, případně stabilitu. Vynecháním některých složek regulátoru (uvažováním příslušných členů nulových) lze získat i jednodušší typy regulátorů, jmenovitě P, PI, PD a I, samotnou D-složku nelze použít – v ustáleném stavu by došlo k rozpojení zpětné vazby (derivace $e(t = \infty)$ je nulová). V následujícím textu bude uvažována paralelní struktura PID regulátoru.



Obr. 1.15 – A – paralelní struktura, B – paralelní struktura s interakcí časových konstant,
C – sériová struktura PID regulátoru

Proporcionální složka

P-složka představuje zesílení zpětné vazby regulátoru a na $e(t)$ reaguje úměrně k její velikosti, větší váha této složky způsobí zrychlení RP může ale také RP destabilizovat, případně lze její vhodnou volbou přivést RP na kmitavou mez stability. Parametry RP, který kmitá s konstantní amplitudou, představují určitou charakteristiku soustavy.

U proporcionálních, na rozdíl od neproporcionálních systémů (systémy s integračním charakterem – jmenovatel obsahuje nulový pól(y)), nezajistí samotná P-složka (případně navíc spolu s D-složkou) dosažení $e(t = \infty) = 0$ – neboť pro nenulovou hodnotu výstupní veličiny je nutný nenulový výstup regulátoru, tedy nenulová regulační odchylka. Teoreticky by bylo možné dosáhnout $e(t = \infty) = 0$ v případě $r_0 = \infty$, z toho vyplývá, že $e(t)$ je tím menší, čím větší je zesílení r_0 .

Integrační složka

I-složka – přidáním nulového pólu prostřednictvím dodatečné I-složky je zajištěna nulová trvalá regulační odchylka URO s proporcionálním systémem. Praktickým příkladem může být zvyšování výkonu topení (integrací $e(t)$ do té doby, než je zajištěn soulad mezi $y(t)$ a $w(t)$). Přidání I-složky však také způsobí zvýšení relativního řádu systému, míry kmitavosti a v důsledku i prodloužení RP. Je také zřejmé, že u integračních soustav není nutné tuto složku přidávat, jelikož nulový pól(y) obsahuje samotná soustava, její přidání

by však mělo význam připadě, že by bylo potřeba sledovat žádanou hodnotu, která by se v čase měnila (nebyla by konstantní) např. průběhu popsateelného směrnicí (rampa) či polynomem vyššího řádu (v takovém případě by bylo potřeba přidat dokonce druhou integrační složku), potom by se dalo hovořit i o poruchách ve smyslu změny zrychlení.

Derivační složka

Na rozdíl od I-složky, má *D-složka* (Honc, 2018) na RP stabilizující vliv. S jejím přidáním se však vážou praktické problémy – D-složka totiž reaguje na rychlé změny s vysokou mírou zesílení (pro skokové změny teoreticky s nekonečným zesílením), přičemž rychlé změny mohou být způsobeny jak změnou řídicí veličiny $w(t)$, tak rychlými změnami regulační odchylky $e(t)$, ty jsou často způsobeny šumem měření – to vede často na nutnost filtrace regulované veličiny. Navíc je tedy bloku derivační složky uvnitř regulátoru představen vnitřní filtr, zde se používá parametr N , ten se volí jako převrácená hodnota časové konstanty filtru – čím je vyšší, tím je míra filtrace nižší a naopak.

S výhodou lze v tomto případě také využít PID regulátoru se dvěma stupni volnosti – váhovými koeficienty b a c u P a D-složky lze snížit reakci na změnu $w(t)$ a v důsledku tak nastavit PID regulátor různěji pro zregulování poruchy.

$$u(t) = r_0(bw(t) - y(t)) + r_{-1} \int_0^t e(\tau) d\tau + r_1 \frac{d(cw(t) - y(t))}{dt}. \quad (64)$$

Kde b je váhový koeficient proporcionální složky PID regulátoru,

c – váhový koeficient derivační složky PID regulátoru.

Téhož efektu lze docílit filtrováním žádané hodnoty. Pokud porucha působí pouze na výstupu soustavy, její zregulování má stejnou dynamiku jako sledování žádané hodnoty a dva stupně volnosti nejsou nutné. Pokud ale působí na vstupu soustavy, nebo má vlastní dynamiku, je zvýšení stupňů volnosti výhodné a regulátor může být naladěn na oba požadavky zvlášť (docílit dobrého sledování $w(t)$ i zregulování poruchy $v(t)$). V praxi se žádaná hodnota často filtruje podobným způsobem jako $e(t)$ pro D-složku regulátoru, nebo se „rampuje“ – při změně $w(t)$ nedojde ke skoku, ale postupně se přejíždí po přímce na požadovanou hodnotu (Honc, 2018).

1.7.1 Požadavky na regulaci

Obecně je cílem regulace vyhovění následujícím kritériím (Cvejn, 2017):

- *Přesnost regulace* – je jí myšleno odstranění trvalé regulační odchylky $e(\infty)$, ta je v případě PID regulátoru dána zesílením regulátoru, počtem nulových pólů ORO (integrační složka zajistí $e(\infty) = 0$ v případě $w = \text{konst.}$ a tvarem trajektorie $w(t)$. U nespojitých regulátorů je $e(t)$ dána hysterezí a dynamikou soustavy (výstup osciluje okolo $w(t)$).
- *Dostatečná rezerva ve stabilitě* – a nemohlo dojít k nestabilitě při změně parametrů soustavy. Lze posuzovat:
 - polohu pólů URO – parametry regulátoru musí být zvoleny takové, aby póly URO ležely v dostatečné vzdálenosti od meze imaginární osy, tj.

$$\Re(s_i) < -\alpha_S, \alpha_S > 0. \quad (65)$$

kde s_i jsou póly URO,

α_S – vzdálenost přímky absolutního útlumu od imaginární osy.

- vzdálenost AFFCh otevřeného regulačního obvodu od kritického bodu – tedy posuzování bezpečnosti v amplitudě a ve fázi.
- *Účinnost regulace* – posuzuje se schopnost:
 - účinně kompenzovat působící poruchy,
 - rychle reagovat na změnu $w(t)$ (problém sledování trajektorie).

Oběma těmito kritériím někdy nelze vyhovět současně, regulační obvod nastavený pro rychlé vyrovnání poruch působících na vstupu soustavy často reaguje podstatně hůře při skokových změnách $w(t)$, případně při poruchách působících na výstupu soustavy $v(t)$. Rezerva ve stabilitě je často uvažována jako prioritní požadavek.

1.8 SYNTÉZA REGULAČNÍHO OBVODU S PID REGULÁTOREM

Možnost ovlivňovat regulační pochod a konečnou hodnotu regulované veličiny patří mezi výhody syntézy regulačních obvodů. Syntézou regulačního obvodu (Cvejn, 2017), (Honc, 2018) rozumíme proces volby regulátoru a jeho seřízení z hlediska požadavků na kvalitu regulace. Nastavením PID regulátoru se rozumí získání nastavitelných parametrů regulátoru, tedy složek proporcionální, integrační a derivační.

Existuje tisíce metod získání nastavitelných parametrů PID regulátoru, účelem této kapitoly je kategorizovat možné přístupy k syntéze regulačního obvodu. Nastavitelné parametry regulátoru lze získat v oblasti časové (např. minimalizací vhodného kritéria), analýzou pólů v komplexní oblasti nebo v oblasti frekvenční. K získání nastavitelných parametrů regulátoru lze přistoupit:

- Empiricky – Experimentální metody vycházejí z URO, praktickou výhodou těchto metod je, že nevyžadují znalost přenosu, princip návrhu spočívá v práci se skutečnou regulovanou soustavou a se skutečným regulátorem. Z tohoto důvodu nevyžadují prakticky žádné znalosti o vlastnostech soustavy. Jsou nejčastěji aplikovány na již existující regulační obvod, které je nutno doladit nebo seřídít po rekonstrukci či opravě, k nejpoužívanějším metodám patří metoda Zieglera a Nicholse, případně metody z ní odvozené (Vítečková M., 2011).

Při empirickém nastavení se zpravidla vyřadí z činnosti integrační složka $T_I \rightarrow \infty$ a derivační složka $T_D \rightarrow 0$, následně je hledána vhodná hodnota k_P , jako druhý se většinou nastavuje I-složka a současně s ní je zpravidla potřeba zpětně korigovat i P-složku, nejinak je tomu i u následného nastavování D-složky, při kterém se koriguje integrační a proporcionální složka regulátoru.

- Analyticky – Pokud je přenos soustavy znám, jsou k dispozici postupy, pomocí kterých je možné dosáhnout kvalitnějších RP. Tyto metody jsou však většinou rovněž komplikovanější, a proto se v praxi méně využívají.

1.8.1 Syntéza v časové oblasti

Do této kategorie můžeme zařadit např. metody seřízení na základě znalosti PCH regulované soustavy, Zieglerovu-Nicholsovou metodu přechodové charakteristiky, seřízení regulátoru podle funkcí standardního tvaru (Whiteleyovy standardní tvary, Naslimova metoda) a seřízení regulátoru podle minima integrálního kritéria (Balátě, 2003).

Nejprve je nutné si definovat pojem *kvalita regulace*. Kvalitu regulace lze posuzovat nejen v časové oblasti (na základě charakteristických rysů přechodového děje URO, nebo vyčíslitelného integrálního kritéria), ale i v oblasti komplexní roviny či v oblasti frekvenční. Pro daný regulační obvod můžeme posuzovat kvalitu regulace nebo provádět syntézu obvodu (volbu regulátoru a jeho seřízení), která požadovanou kvalitu regulace zajistí.

Co se posuzování kvality ve smyslu charakteristických rysů přechodového děje týče, tak v časové oblasti je posuzována odezva URO na různé typy standardizovaných testovacích signálů. Tento postup je výhodný, protože existuje korelace mezi odezvou na standardní signál a schopností obvodu fungovat v obvyklých provozních podmínkách – průběh vstupní veličiny je totiž podobný pro mnoho procesů v průmyslu. Navíc je použitím standardních testovacích signálů umožněno obecně porovnávat více různých návrhů.

Mezi standardní testovací signály patří jednotkový (Heavisideův) skok, dále pak tzv. rampa a jednotkový (Diracův) impuls. Následující platí při posuzování chování URO při skokové změně řídicí veličiny (jednotkový skok).

Z hlediska posouzení jakosti RP rozlišujeme (Richard C. Dorf, 2016):

- *Doba odezvy* t_0 – Pokud je však systém přetlumený $\zeta \geq 1$, tak t_{\max} není definována, místo toho je potom použita modifikovaná doba odezvy t_{0m} , která představuje dobu od dosažení 10 % do 90 % $y(\infty)$.
- t_{\max} – doba dosažení y_{\max} .
- *Relativní překmit* μ – je užitečný pro netlumené systémy $\zeta < 1$ s překmitem, je definován jako

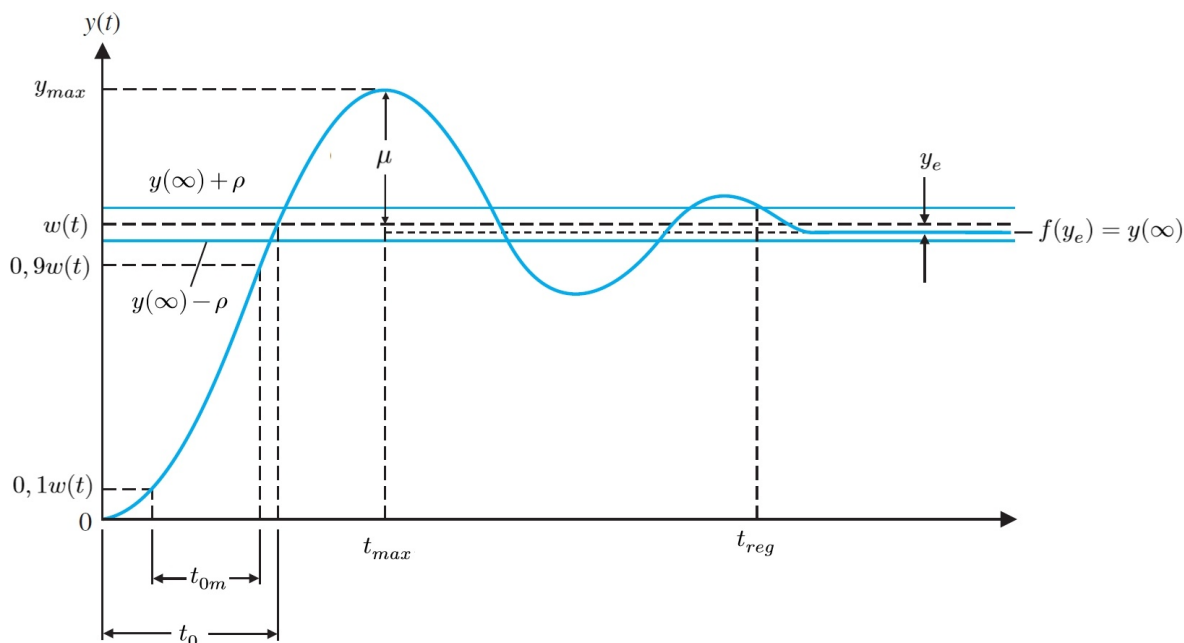
$$\mu = \frac{y_{\max} - y_{\infty}}{y_{\infty}}. \quad (66)$$

Kde y_{\max} je nejvyšší hodnota regulované veličiny,

y_{∞} – ustálená hodnota regulované veličiny.

Uvedené parametry charakterizují rychlost odezvy URO.

- *Doba regulace* t_{reg} – doba kdy $y(t)$ poprvé vstoupí do určitého procentuálně vyjádřeného pásma $\rho = \delta y_{\infty}$ a již ho neopustí, přičemž obvyklá hodnota parametru je $\delta = 0,01 \div 0,05$.



Obr. 1.16 – Ukazatele kvality RP v časové oblasti, (Richard C. Dorf, 2016)

Posuzovat kvalitu v časové oblasti lze i pomocí kvantitativních *integrálních kritérií*, ty umožňují přesné vyčíslení jedné hodnoty, která souhrnně charakterizuje míru kvality daného regulačního pochodu. Typicky je toto kritérium minimalizováno (analytickou či numerickou metodou) prostřednictvím změny nastavitelných parametrů regulátoru. Potom lze hovořit o optimálním regulátoru z pohledu daného kritéria. Obecný tvar integrálního kritéria je

$$J = \int_0^t f(w(\tau), y(\tau), e(\tau), \tau) d(\tau). \quad (67)$$

Různou kombinací argumentů funkce lze obdržet různá integrální kritéria (Richard C. Dorf, 2016).

Mezi základní kvantitativní integrální kritéria patří (Richard C. Dorf, 2016), (Honc, 2018):

- Lineární regulační plocha (IE – integral of error) je definována jako

$$J_{IE} = \int_0^t e(\tau) d\tau. \quad (68)$$

- Absolutní regulační plocha (IAE – integral of absolute error) je definována jako

$$J_{IAE} = \int_0^t |e(\tau)| d\tau. \quad (69)$$

Kritérium je vhodné pro kmitavé i nekmitavé RP, pro kmitavé RP jej však nelze spočítat analyticky, pouze simulačně (numericky).

- Kvadratická regulační plocha (ISE – integral of squared error) je definována jako

$$J_{\text{ISE}} = \int_0^t e^2(\tau) d\tau. \quad (70)$$

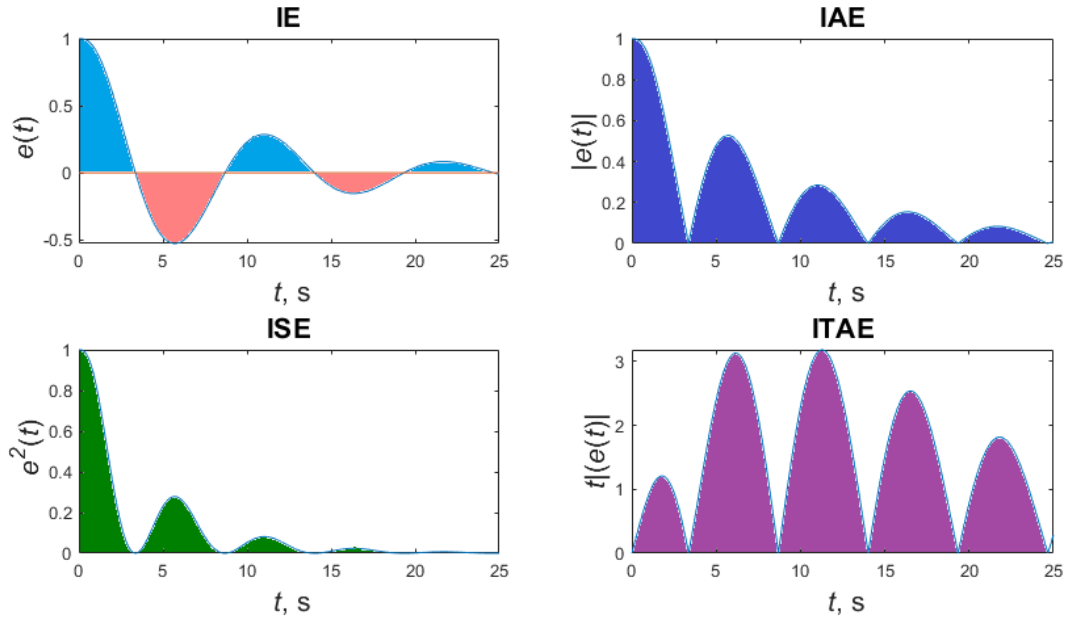
Kritérium je vhodné pro kmitavé i nekmitavé RP, pro kmitavé RP oproti IAE jej však lze spočítat analyticky, nevýhodou je, že výsledné (resp. nalezené optimální nastavitelné parametry regulátoru) RP bývají až příliš kmitavé. Tento problém lze odstranit rozšířením kritéria, ve smyslu uvažování derivace $y(t)$ (viz dále). Toto kritérium bude navíc rozlišovat mezi podtlumeným a přetlumeným systémem (URO), minimální hodnota tohoto kritéria bude odpovídat kompromisní hodnotě relativního tlumení.

- Absolutní regulační plocha násobená časem (ITAE – integral of time multiplied by an absolute error) zahrnuje čas i regulační odchylku a proto při minimalizaci tohoto kritéria dochází současně k minimalizaci absolutní regulační plochy a doby regulace.

$$J_{\text{ITAE}} = \int_0^t t|e(\tau)| d\tau. \quad (71)$$

Kritérium je vhodné pro kmitavé i nekmitavé RP, pro kmitavé RP jej však nelze spočítat analyticky, ale pouze simulačně (numericky), stejně jako IAE.

Průběhy jednotlivých regulačních odchylek pro kritéria IE, IAE, ISE a ITAE jsou spolu s příslušnou regulační plochou znázorněny na obr. 1.17.



Obr. 1.17 – Průběh regulační odchylky jednotlivých integrálních kritérií (68), (69), (70), (71) a příslušná regulační plocha

Modifikace výpočtu kvadratické regulační plochy

Aby výsledky výpočtu kvadratické plochy *numerickou metodou* byly porovnatelné, musí se návrhář zabývat *konečným horizontem řízení*, tedy okamžiku ukončení výpočtu, ten totiž bude pro každý systém jiný. Časový horizont by měl být zvolen tak, aby se regulační odchylka dostatečně přiblížila nule, jinak je výpočet zkreslený. U kmitavých systémů může nastat situace, kdy bude $e(t) = 0$ i když RP nebude ustálený, u procesů blízko meze stability může být navíc tento časový horizont velmi dlouhý. Jako konečný horizont řízení je vhodné volit časový úsek odpovídající době regulace $t = t_{reg}$, tedy

$$J_{ISE} = \int_0^{t=t_{reg}} e^2(\tau) d\tau. \quad (72)$$

Alternativně lze využít *analytickou metodu* výpočtu kvadratické regulační plochy. Definiční vztah minimalizace kvadratické regulační plochy lze zapsat jako (Balátě, 2003)

$$J_{ISE} = \int_0^t (w(\tau) - y(\tau))^2 d\tau = \int_0^t (y(\infty) - y(\tau))^2 d\tau = J_{ISE}(k_P, k_I, k_D) \rightarrow \min. \quad (73)$$

Výpočet integrálu (73) lze provést přímo, nebo se převede Fourierovou transformací na integrál z kvadrátu modulu spektra (s proměnným kmitočtem ω) – na tzv. Parsevalův integrál P

$$P(k_P, k_I, k_D) = \int_0^\infty (w(\tau) - y(\tau))^2 d\tau = \frac{1}{2\pi i} \int_0^\infty |E(i\omega)|^2 d i\omega = \frac{1}{\pi} \int_0^\infty |E(i\omega)|^2 d\omega, \quad (74)$$

kde $E(i\omega)$ je Fourierův obraz (spektrum) regulační odchyly $e(t)$, $|E(i\omega)|^2 = E(i\omega)E(-i\omega)$ (Balátě, 2003).

Dále je uvažován Laplaceův obraz regulační odchyly $e(t)$, který je vyjádřen racionální funkcí lomenou

$$E(s) = Y(s) - \frac{1}{s}y(\infty) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} \quad (75)$$

Po provedení integrace (75) je pro ISE obdržén obecný výraz, který lze upravit na tvar

$$J_{\text{ISE}} = P = \frac{1}{2a_n} \frac{H_1}{H}, \quad (76)$$

kde H a H_1 jsou determinanty sestavené z koeficientů obrazu regulační odchyly (75) upraveného na tvar

$$E(s) = \frac{b_{n-1} s^{n-1} + \dots + b_1 s + b_0}{a_n s^n + \dots + a_1 s + a_0}, \quad (77)$$

a sice H je Hurwitzův determinant (viz pododíl 1.6.1) n -tého stupně sestavený z koeficientů jmenovatele (77), H_1 vznikne z H , nahradíme-li jeho první řádek prvky (78)

$$\begin{aligned} q_0 &= (-1)^0 b_{n-1}^2, \\ q_1 &= (-1)^1 (b_{n-2}^2 - 2b_{n-1}b_{n-3}), \\ q_2 &= (-1)^2 (b_{n-3}^2 - 2b_{n-2}b_{n-4} + 2b_{n-1}b_{n-5}), \\ &\vdots \\ q_{n-3} &= (-1)^{n-3} (b_2^2 - 2b_1b_3 + 2b_0b_4), \\ q_{n-2} &= (-1)^{n-2} (b_1^2 - 2b_0b_2), \\ q_{n-1} &= (-1)^{n-1} b_0^2, \end{aligned} \quad (78)$$

a tedy q_0 je prvek prvního a q_{n-1} prvkem posledního sloupce determinantu H_1 (80), takže

$$H = \begin{bmatrix} a_{n-1} & a_{n-3} & 0 \\ a_n & a_{n-2} & 0 \\ 0 & 0 & a_n \end{bmatrix}, \quad (79)$$

$$H_1 = \begin{bmatrix} q_0 & q_1 & q_{n-1} \\ a_n & a_{n-2} & 0 \\ 0 & 0 & a_0 \end{bmatrix}. \quad (80)$$

Hodnoty nastavitelných parametrů regulátoru lze získat z podmínek, že parciální derivace (Balátě, 2003)

$$\frac{\partial J_{\text{ISE}}}{\partial k_{(\text{P,I,D})}} = 0. \quad (81)$$

Vztah (76) platí pouze pro stabilní RP, je tedy hledáno minimum funkce P , které leží uvnitř oblasti stability regulačního obvodu. Hodnota P může být záporná v případě, že daná sada nastavitelných parametrů regulátoru vede na nestabilní URO (Balátě, 2003).

Nalezené nastavitelné parametry regulátoru odpovídající nalezenému extrému regulační plochy vykazují sice malé regulační odchylky, na druhou stranu však vedou k RP s vyšší mírou kmitavosti (oproti kritériu minima lineární regulační plochy). Pro snížení míry kmitavosti je možné kritérium (73) rozšířit o derivaci $y(t)$ (Balátě, 2003)

$$J_{\text{ISE}} = \int_0^t \left[(y(\infty) - y(\tau))^2 + \kappa \left(\frac{dy}{d\tau} \right)^2 \right] d\tau = J_{\text{ISE}}(k_P, k_I, k_D) \rightarrow \min. \quad (82)$$

1.8.2 Návrh dle umístění pólů přenosu

V principu se jedná o posuzování kvality RP na základě stupně stability. Řešení charakteristické rovnice URO (tj obecný integrál vlastních kmitů regulačního obvodu) je obecně ve tvaru

$$y(t) = \sum_{i=1}^n C_i s_i^{s_i t}, \quad (i = 1, 2, 3, \dots n) \quad (83)$$

kde $s_{i,i+1} = \alpha_i + j\omega_i$ jsou nenásobné kořeny charakteristické rovnice URO.

Reálné kořeny $s_i = -\alpha_i$ určují stabilní aperiodické složky řešení

$$y_i(t) = C_i e^{-\alpha_i t}. \quad (84)$$

Komplexně sdružené kořeny $s_{i,i+1} = \alpha_i + j\omega_i$ určují stabilní kmitavé složky řešení (Balátě, 2003)

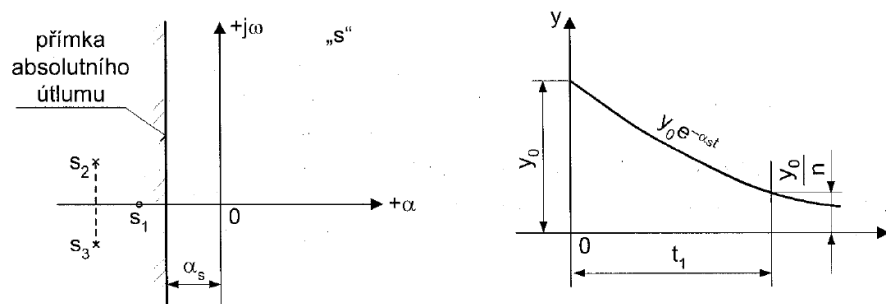
$$y_i(t) = C_{i1} e^{(-\alpha_i + j\omega_i)t} + C_{i2} e^{(-\alpha_i - j\omega_i)t}. \quad (85)$$

Obvod je tím stabilnější, čím více jsou kořeny charakteristické rovnice URO vzdáleny doleva od imaginární osy roviny kořenů s s tím, že rychlost konvergence přechodné $e(t)$ bude tím vyšší, čím je tato vzdálenost větší. Současně bude také větší bezpečnost ve stabilitě – tímto způsobem lze definovat stupeň stability jako vzdálenost $|\alpha_s|$, o kterou je posunuta přímka absolutního útlumu vlevo od imaginární osy v rovině s , ta současně určuje velikost tlumení. Míra tlumení se zvyšuje spolu s růstem hodnoty $|\alpha_s|$. Velikost tlumení lze určit z podmínky, že za dobu t_1 bude počáteční odchylka y_0 n -krát menší. Z toho plyne, že všechny kořeny charakteristické rovnice musí ležet vlevo od *přímky absolutního útlumu*, která leží ve vzdálenosti $|\alpha_s|$ rovnoběžně s imaginární osou. Úlohu lze převést na řešení stability pro posunutý souřadnicový systém (nalevo o hodnotu $|\alpha_s|$) (Balátě, 2003).

Platí, že u kmitavých složek $y(t)$ můžeme vypočítat pokles amplitudy za dobu jednoho kmitu

$$A_2 = A_1^{-\alpha_i T_i}, e^{-2\pi \frac{\alpha_i}{\omega_i}}, \quad (86)$$

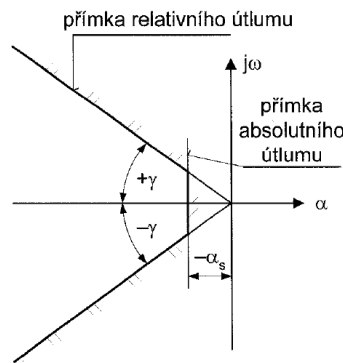
s tím, že čím vyšší bude u komplexně sdružených kořenů $s_{i,i+1} = \alpha_i \pm j\omega_i$ větší poměr reálné složky k imaginární, tím rychleji se bude amplituda kmitavé složky zmenšovat (Balátě, 2003).



Obr. 1.18 – Přímka absolutního útlumu (Balátě, 2003)

V rovině kořenů lze kromě přímky absolutního útlumu (oblast kořenů s minimálním stupněm stability $|\alpha_s|$ vymežit i oblasti s minimálním logaritmickým dekrementem útlumu – *přímka relativního útlumu* $\omega = \tan(\gamma\alpha)$, ta je vymezena úhly $\pm\gamma$, přičemž $\gamma = \arctan(\frac{\omega}{\alpha})$. Póly kmitavých složek, které mají stejný součinitel relativního tlumení leží na těchto přímkách symetricky vycházejících z počátku (Balátě, 2003).

Požadovaný průběh přechodového děje lze vymežit oblastí rozložení kořenů charakteristické rovnice, přičemž rozhodující vliv na průběh RP budou mít ty kořeny charakteristické rovnice, jejichž hodnoty $|\alpha| > |\alpha_s|$ jsou hodnotě $-\alpha_s$ nejbližší – dominantní kořeny (Balátě, 2003).



Obr. 1.19 – Oblast rozložení kořenů charakterizována k požadovanému průběhu přechodového děje (Balátě, 2003)

1.8.3 Syntéza ve frekvenční oblasti

Metoda optimálního modulu

Do syntézy ve frekvenční oblasti lze zařadit metodu optimálního modulu. Ta vychází z přenosu řízení URO $G_w(s)$, v ideálním případě by měl být splněn požadavek

$$G_w(s) = 1, \quad (87)$$

tuto podmínku lze ve frekvenční oblasti vyjádřit jako

$$G_w(s) = 1 \Rightarrow |G_w(i\omega)| = A_w(\omega) = 1, \quad (88)$$

v reálných podmínkách bude platit

$$A_w(\omega) \rightarrow 1, \quad (89)$$

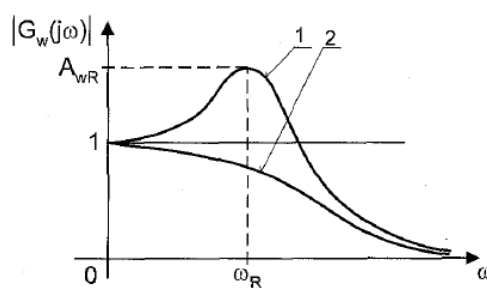
přičemž $A_w(\omega)$ je modul (amplituda) frekvenčního přenosu řízení URO. Tomuto požadavku odpovídají průběhy modulu přenosů řízení URO

$$|G_w(i\omega)| = f(\omega), \quad (90)$$

znázorněné na obr. 1.20, přičemž za nevhodný průběh je považovaný takový, který má při úhlovém kmitočtu ω_R rezonanční převýšení

$$A_{wR}(\omega_R) > 1, \quad (91)$$

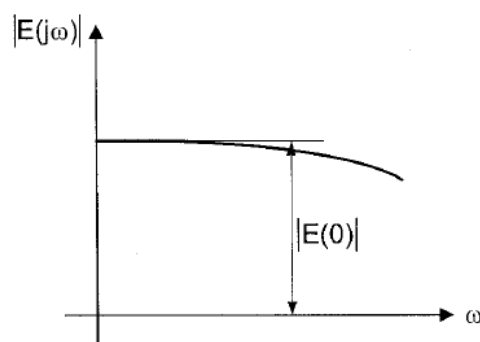
což by způsobilo velké překmity PCH a tedy rezonanční jev, tj. náchylnost URO k málo tlumeným kmitům. Existuje tedy souvislost mezi rezonančním zvětšením modulu přenosu řízení URO a relativním překmitem (Balátě, 2003).



Obr. 1.20 – Průběhy modulu přenosu řízení URO, 1 – nevhodný průběh, 2 – požadovaný průběh (Balátě, 2003)

Za optimální průběh modulu $A_w(\omega)$ tj. AFCh uzavřeného regulačního obvodu, je třeba požadovat průběh, který je monotónně klesající funkcí úhlového kmitočtu ω , tj. aby pro (90) platilo (Balátě, 2003)

$$\frac{d|G_w(i\omega)|}{d\omega} \leq 0. \quad (92)$$



Obr. 1.21 – Požadovaný průběh modulu přenosu řízení URO (Balátě, 2003)

Totéž co platí pro $|G_{i\omega}(s)|$, musí platit i pro jeho druhou mocninu

$$|G_w(i\omega)| \rightarrow 1 \Rightarrow |G_w(i\omega)|^2 \rightarrow 1. \quad (93)$$

Význam je v tom, že s druhou mocninou se lépe pracuje – v důsledku se nepracuje s odmocninou a snadněji se také určí.

Pro součin komplexně sdružených čísel platí

$$(\alpha + i\omega)(\alpha - i\omega) = \alpha^2 + \omega^2 = |\alpha + i\omega|^2 \quad (94)$$

a tedy lze psát

$$|G_w(i\omega)|^2 = A_w^2(\omega) = G_w(i\omega)G_w(-i\omega) \quad (95)$$

Předpokládá-li se přenos $G_w(s)$ ve tvaru

$$G_w(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}, \quad n \geq m. \quad (96)$$

Potom druhá mocnina modulu řízení, v kterém se vyskytují pouze sudé mocniny ω , má tvar (Balátě, 2003)

$$\begin{aligned} |G_w(i\omega)|^2 &= G_w(i\omega)G_w(-i\omega) = \\ &= \frac{B_m \omega^{2m} + B_{m-1} \omega^{2(m-1)} + \dots + B_1 \omega^2 + B_0}{A_n \omega^{2n} + A_{n-1} \omega^{2(n-1)} + \dots + A_1 \omega^2 + A_0}, \end{aligned} \quad (97)$$

kde

$$\begin{aligned} B_0 &= b_0^2, \\ B_1 &= b_1^2 - 2b_0 b_2, \\ B_2 &= b_2^2 - 2b_1 b_3 + 2b_0 b_4, \\ &\vdots \\ &\vdots \\ B_{m-1} &= b_{m-1}^2 - 2b_{m-2} b_m, \\ B_m &= b_m^2, \end{aligned} \quad (98)$$

a

$$\begin{aligned}
A_0 &= a_0^2, \\
A_1 &= a_1^2 - 2a_0a_2, \\
A_2 &= a_2^2 - 2a_1a_3 + 2a_0a_4, \\
A_3 &= a_3^2 - 2a_2a_4 + 2a_1a_5 - 2a_0a_6, \\
&\vdots \\
A_{n-1} &= a_{n-1}^2 - 2a_{n-2}a_n, \\
A_n &= a_n^2.
\end{aligned} \tag{99}$$

Úplné nezávislosti $|G_w(i\omega)|$ URO na kmitočtu (90), resp. $|G_w(i\omega)|^2$ bychom dosáhli, kdyby stupeň polynomu čitatele byl roven stupni jmenovatele, tato podmínka však nebývá splněna (je nereálná). Pro vyhovující průběh RP stačí, aby byla splněna podmínka monotónního průběhu funkce $|G_w(i\omega)|^2$, tj.

$$\frac{B_0}{A_0} \geq \frac{B_k}{A_k} \Rightarrow A_k B_0 = A_0 B_k, \tag{100}$$

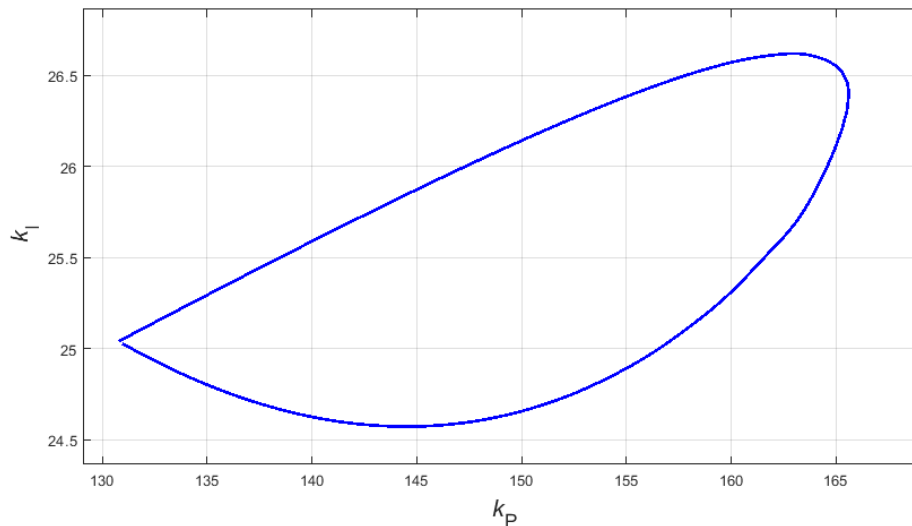
kde k může nabýt hodnoty $k = 1, 2, \dots, \alpha$, kde α je počet nastavitelných parametrů regulátoru (Balátě, 2003).

Tímto způsobem je získána soustava podmínek, jejichž řešením může být dáno optimální seřízení regulátoru, součinitele A_i, B_i ve vztahu (97) jsou funkcemi nastavitelných parametrů regulátoru, stejně tak i a_i, b_i ve vztahu (96), přičemž relace mezi nimi tvoří (98) resp. (99). Řešením těchto vztahů jsou získány hodnoty nastavitelné parametry regulátoru (Balátě, 2003).

Je třeba zdůraznit, že kritérium metody optimálního modulu v sobě nezahrnuje podmínky stability, a proto je nutné splnění podmínek stability vždy ověřit (Balátě, 2003).

2 NÁVRH PI(D) REGULÁTORU NA ZÁKLADĚ SESTROJENÍ OBLASTI STABILITY

Oddíl 1.8 se zabýval získáním nastavitelných parametrů regulátoru, konkrétněji jedné sady nastavitelných parametrů regulátoru – jednoho bodu v rovině parametrů, který z pohledu dané metody představuje optimální nastavení. Existuje však samostatná oblast výzkumu, která se zabývá nalezením celé množiny přípustných nastavení. V tomto případě je, z hlediska stability, výsledkem množina stabilizujících nastavitelných parametrů regulátoru – tzv. oblast stability (viz obr. 2.1).



Obr. 2.1 – Příklad oblasti stability v rovině (k_P, k_I)

Získání oblasti stability je velice výhodné, lze ji využít např. k nalezení optimálního nastavení pomocí optimalizace – např. s využitím integrálního kritéria (viz oddíl 1.8.1). Výsledkem je bod v rovině (k_P, k_I) , případně bod v prostoru parametrů (k_P, k_I, k_D) , který se nachází uvnitř oblasti stability a představuje optimální nastavení. V takovém případě lze hovořit o návrhu PI(D) regulátoru na základě sestrojení oblasti stability.

Oblast stability lze však dále využít např. k robustnímu řízení. Získaná oblast stability poskytuje také velice užitečnou informaci v případě nestabilních systémů, pro které může být empirické hledání stabilizujících parametrů velice pracné a nalezené stabilizující nastavení nemusí být zdaleka optimální.

V posledních desetiletích byla v odborné literatuře popsána řada metod, které se konstrukcí oblasti stability pro lineární systémy zabývají. Mezi moderní metody konstrukce

oblasti stability patří tzv. *metoda hraniční křivky* (dále HK) (Tan et al., 2006), kterou se tato práce zabývá. Kromě metody HK je vhodné zmínit také přístup spočívající v hledání singulárních frekvencí a zobecněné Hermitově-Biehlerově větě (Bajcinca, 2006), (Silva et al., 2001).

Ke konstrukci oblasti stability na základě Hermitovy-Biehlerovy věty je možné využít nástroje lineárního programování. Jedná se o efektivní přístup, které mimo jiné odhalil důležité strukturální vlastnosti PI(D) regulátorů, např., že pro konstantní k_P leží stabilizující parametry integrační a derivační složky uvnitř konvexní množiny. Nicméně výpočetní náročnost tohoto přístupu narůstá exponenciálně s řádem uvažovaného systému. Další nevýhodou přístupu založeném na lineárním programování je nutnost prohledávání v určitém rozsahu k_P .

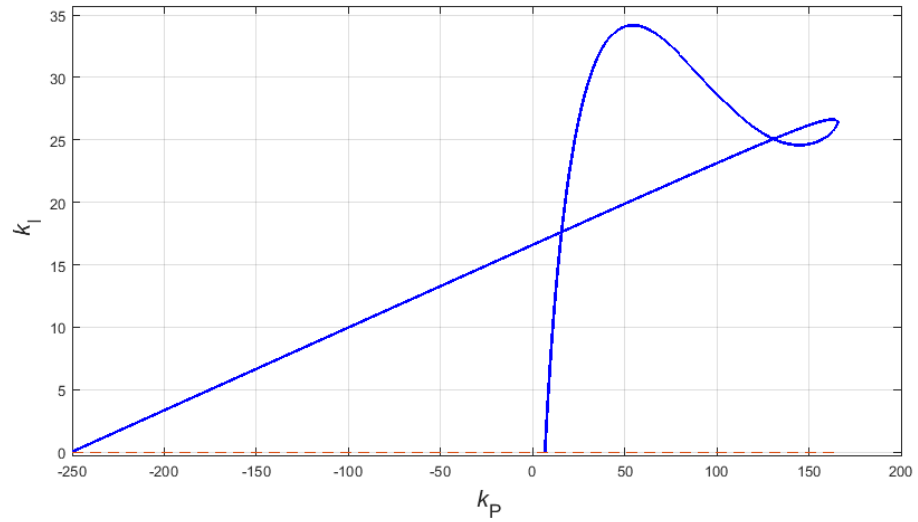
Alternativou k hledání oblasti stability pomocí lineárního programování je přístup založený na frekvenčních charakteristikách (Tan et al., 2006).

2.1 METODA HRANIČNÍ KŘIVKY

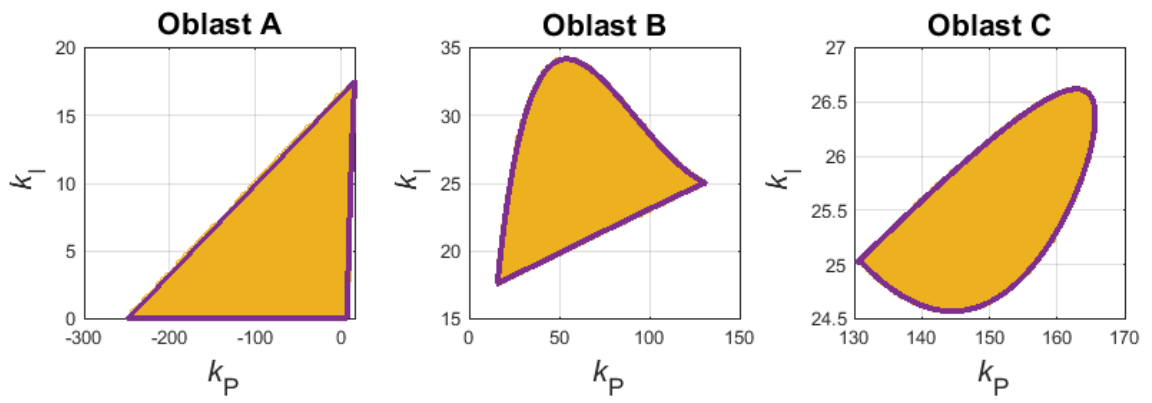
Tato práce se zabývá praktickou implementací metodou hraniční křivky, která využívá tzv. D-dekompozice a umožňuje získání oblasti stability. Základem realizace je článek (Tan et al., 2006), kde je sice princip této metody popsán, ale nejsou zde popsány detaily, které je nutné řešit při softwarové realizaci. Postup numerické metody hraniční křivky je založený na výpočtu hraniční křivky (angl. Stability Boundary Locus).

HK je funkce $(k_P, k_I) = l(\omega)$. Jedná se o parametricky vyjádřenou křivku, dosazením ω do $k_P(\omega)$, $k_I(\omega)$ lze získat body v rovině (k_P, k_I) a ty následně graficky znázornit. Příklad průběhu HK v rovině parametrů PI regulátoru (k_P, k_I) je vidět na obr. 2.2.

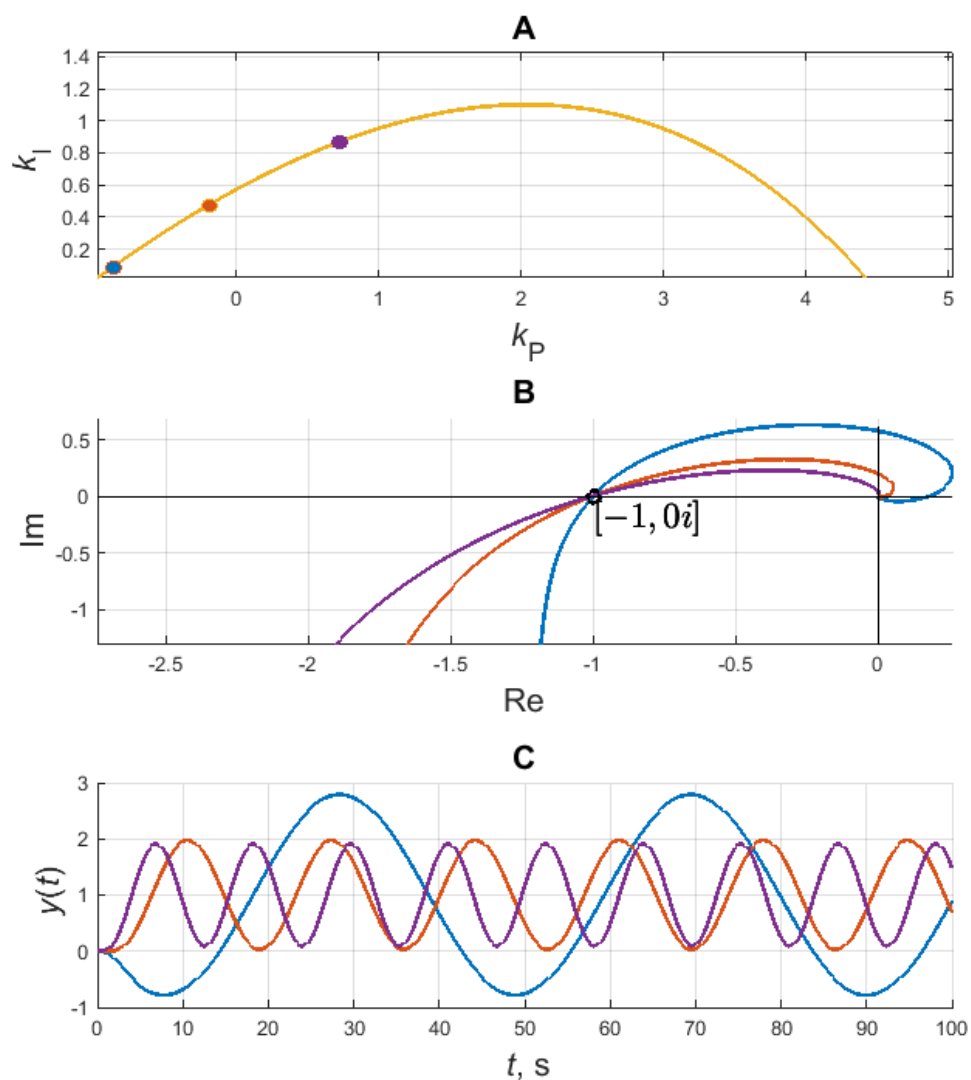
HK svým průběhem vymezení roviny parametrů na jednotlivé oblasti s tím, že jednotlivé oblasti (viz obr. 2.3) mohou být buď stabilní či nestabilní. V případě, že je daná oblast stabilní, tak je možné na segmenty HK, které danou oblast vymezují, nahlížet jako na mez stability – nastavitelné parametry regulátoru, ležící na hranici oblasti, přivedou URO na kmitavou mez stability viz obr. 2.4.



Obr. 2.2 – Příklad průběhu hraniční křivky



Obr. 2.3 – Oblasti vymezené průběhem HK



Obr. 2.4 – Ukázka demonstrující segment průběhu HK jako sadu nastavení, které vedou na kmitavou mez stability. A – výběr tří sad nastavitelných parametrů regulátoru, B – AFFCh otevřeného regulačního obvodu odpovídající vybraným 3 sadám nastavení, C – regulační pochody získané pomocí vybraných 3 sad nastavení

Pro výpočet hraniční křivky předpokládejme paralelní strukturu regulátoru

$$C(s) = k_P + \frac{k_I}{s} = \frac{k_P s + k_I}{s}, \quad (101)$$

a systém s jedním vstupem a jedním výstupem

$$G(s) = \frac{N(s)}{D(s)}. \quad (102)$$

Úkolem je vypočítat (vykreslit) průběh HK, která může (ale *nemusí*) svým průběhem vymezit oblast stabilních parametrů regulátoru (101). Nejprve je vyjádřen frekvenční

přenos systému a následně je provedeno rozložení na sudou (index e) a lichou (index o) část

$$G(j\omega) = \frac{N(j\omega)}{D(j\omega)} = \frac{N_e(-\omega^2) + j\omega N_o(-\omega^2)}{D_e(j\omega) + j\omega D_o(-\omega^2)}, \quad (103)$$

charakteristický polynom URO má tvar

$$\begin{aligned} \Delta(s) = & [k_I N_e(-\omega^2) - k_P \omega^2 N_o(-\omega^2) + -\omega^2 D_o(-\omega^2)] + \\ & + j [k_P \omega N_e(-\omega^2) + k_I \omega N_o(-\omega^2) + \omega D_e(-\omega^2)]. \end{aligned} \quad (104)$$

Položením reálné a imaginární části $\Delta(s)$ rovno nule lze získat

$$\begin{aligned} k_P(-\omega^2 N_o(-\omega^2)) + k_I(N_e(-\omega^2)) &= \omega^2 D_o(-\omega^2); \\ k_P(\omega N_e(-\omega^2)) + k_I(\omega N_o(-\omega^2)) &= -\omega D_e(-\omega^2), \end{aligned} \quad (105)$$

kde lze jednotlivé členy substituovat

$$\begin{aligned} Q(\omega) &= -\omega^2 N_o(-\omega^2); \quad R(\omega) = N_e(-\omega^2); \\ S(\omega) &= \omega N_e(-\omega^2); \quad U(\omega) = \omega N_o(-\omega^2); \\ X(\omega) &= \omega^2 D_o(-\omega^2); \quad Y(\omega) = -\omega D_e(-\omega^2). \end{aligned} \quad (106)$$

Rovnice

$$\begin{aligned} k_P Q(\omega) + k_I R(\omega) &= X(\omega), \\ k_P S(\omega) + k_I U(\omega) &= Y(\omega), \end{aligned} \quad (107)$$

lze upravit na tvar

$$\begin{aligned} k_P &= \frac{X(\omega)U(\omega) - Y(\omega)R(\omega)}{Q(\omega)U(\omega) - R(\omega)S(\omega)}, \\ k_I &= \frac{Y(\omega)Q(\omega) - X(\omega)S(\omega)}{Q(\omega)U(\omega) - R(\omega)S(\omega)}, \end{aligned} \quad (108)$$

který představuje parametrické vztahy, které umožňují vykreslit průběh HK $l(\omega)$ v rovině (k_P, k_I) (Tan et al., 2006).

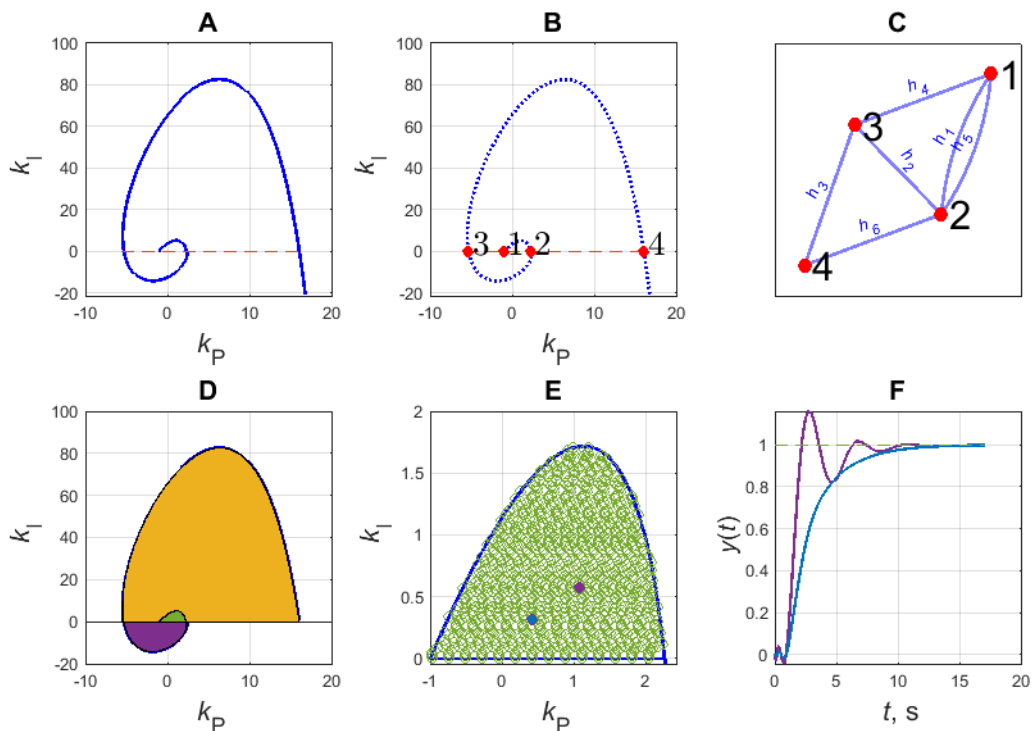
Významnou výhodou metody hraniční křivky oproti ostatním přístupům uvedených výše je, že umožňuje nalézt oblast stability na menším rozsahu frekvencí, tím je tato metoda efektivní z hlediska výpočetního času. Metoda je navíc rozšířitelná ve smyslu požadavku na určitou rezervu ve stabilitě či k zajištění specifikované bezpečnosti v amplitudě a ve

fázi. Dále je metoda rozšiřitelná o parametr derivační složky k_D , či pro výpočet oblasti stability systému s neurčitými parametry (Tan et al., 2006).

Obecný princip získání stabilní oblasti lze popsat následujícími kroky:

1. *Výpočet průběhu HK* (viz obr. 2.5 – A) – pomocí (108) je proveden výpočet HK pro zvyšující se hodnotu ω , tímto se zabývá pododíl 2.2.1. Zároveň je v rámci tohoto kroku potřeba řešit okamžik ukončení výpočtu (viz pododíl 2.2.2) a s ním související úpravu přenosu ve smyslu jeho normalizace na kladné zesílení (16), případně jeho rozšíření vhodným dynamickým členem v rámci zajištění korektního průběhu HK (viz dále v pododílu 2.2.3).
2. *Nalezení významných průsečíků HK* (viz obr. 2.5 – B) – jelikož je snahou individuální reprezentace oblastí, tak je potřeba tyto oblasti nějakým způsobem z průběhu HK extrahovat. K tomu lze využít významných bodů, které představují průsečíky křivky se sebou samou, a dále také průsečíky HK s osou $k_I = 0$. Úloha hledání průsečíků je v podstatě úloha hledání indexů v poli (k_P, k_I) , ve kterých dochází k průsečíku. Problematika hledání průsečíků je podrobně popsána v pododílu 2.2.4
3. *Reprezentace průběhu HK jako grafu* – indexy v poli popisujícím průběh HK, které odpovídají významným průsečíkům získaným v předchozím kroku v podstatě oddělují jednotlivé segmenty HK. Průběh HK lze tedy reprezentovat jako graf, kde průsečíky představují vrcholy a segmenty HK mezi nimi představují hrany (viz obr. 2.5 – C). Problematika konstrukce grafů je podrobně popsána v pododílu 2.2.5.
4. *Individuální reprezentace jednotlivých oblastí a zajištění jejich správného uspořádání z hlediska topologie* – v této části je v získaném grafu provedeno nalezení tzv. elementárních smyček grafu, které představují posloupnost hran resp. vrcholů, které definují jednotlivé oblasti, jejich reprezentací se podrobně zabývá pododíl 2.2.6. Jelikož však graf neobsahuje kompletní informaci o vzájemném uspořádání oblastí, tak je potřeba provést ještě tzv. topologické uspořádání oblastí, které má zajistit, že vnitřní oblast, která se nachází celá uvnitř vnější jiné, bude při zobrazení vnější oblasti vyjmuta (např. oblast mezi vrcholy 1 a 2 na obr. 2.5 – B, je celá obklopena jinou vnější oblastí). Situaci po tomto kroku zachycuje obr. 2.5 – D, kde jsou jednotlivé oblasti znázorněny barevně. Problematikou topologického uspořádání oblastí se podrobně zabývá pododíl 2.2.7.

5. *Vyplnění oblastí sítí bodů* (viz obr. 2.5 – E) – po předchozích krocích je oblast reprezentována prostřednictvím svých hraničních segmentů, pro další práci s oblastmi je však potřeba reprezentovat také jejich vnitřní prostor. Problematikou nalezení bodů uvnitř oblasti se zabývá pododdlíl 2.2.8.
6. *Nalezení stabilních oblastí a optimálních nastavení v jejich vnitřním prostoru* – Pro každou samostatně reprezentovanou oblast je určen vnitřní bod (resp. body), v dostatečné vzdálenosti od hranice oblasti. Následně je určeno, zda-li je URO pro nalezený bod (k_P, k_I) stabilní a pokud ano, tak je jako stabilní označena celá oblast. Pro oblasti označené jako stabilní je následně provedena optimalizace, která spočívá v hledání optimálního nastavení z pohledu daného integrálního kritéria. Pro takto nalezenou sadu optimálních nastavitelných parametrů (viz obr. 2.5 – E) je následně vykreslen odpovídající regulační pochod (viz obr. 2.5 – F). Problematika nalezení stabilních oblastí a optimálního nastavení je podrobně popsána v pododdlílu 2.2.9.



Obr. 2.5 – Ilustrace zachycující obecné kroky získání stabilní oblasti a vnitřního bodu reprezentujícího optimální nastavení

2.2 REALIZACE METODY HRANIČNÍ KŘIVKY

V této části jsou detailně popsány kroky realizace postupu popsaného v oddílu 2.1. Realizace metody hraniční křivky spočívá v algoritmizaci postupu, uvedeného v oddílu 2.1 s cílem automatické konstrukce oblasti stability pro lineární systémy s jedním vstupem a výstupem bez dopravního zpoždění. Získaná oblast stability je následně využita k nalezení optimálního nastavení pomocí vhodného integrálního kritéria, přičemž volba tohoto kritéria je dále diskutována.

Následující jednotlivé pododdíly tohoto oddílu představují kompaktní logické celky, které podrobně popisují řešenou problematiku. Pododdíly jsou chronologicky seřazené a popisují funkci celého algoritmu včetně komplikací, které bylo v rámci implementace potřeba řešit. Dále diskutují přístupy k řešení daného podproblému, které nejsou z hlediska řešení vhodné či výhodné a je v nich prezentován nejvhodnější nalezený přístup. Součástí je také popis jednotlivých algoritmů ve formě pseudokódu.

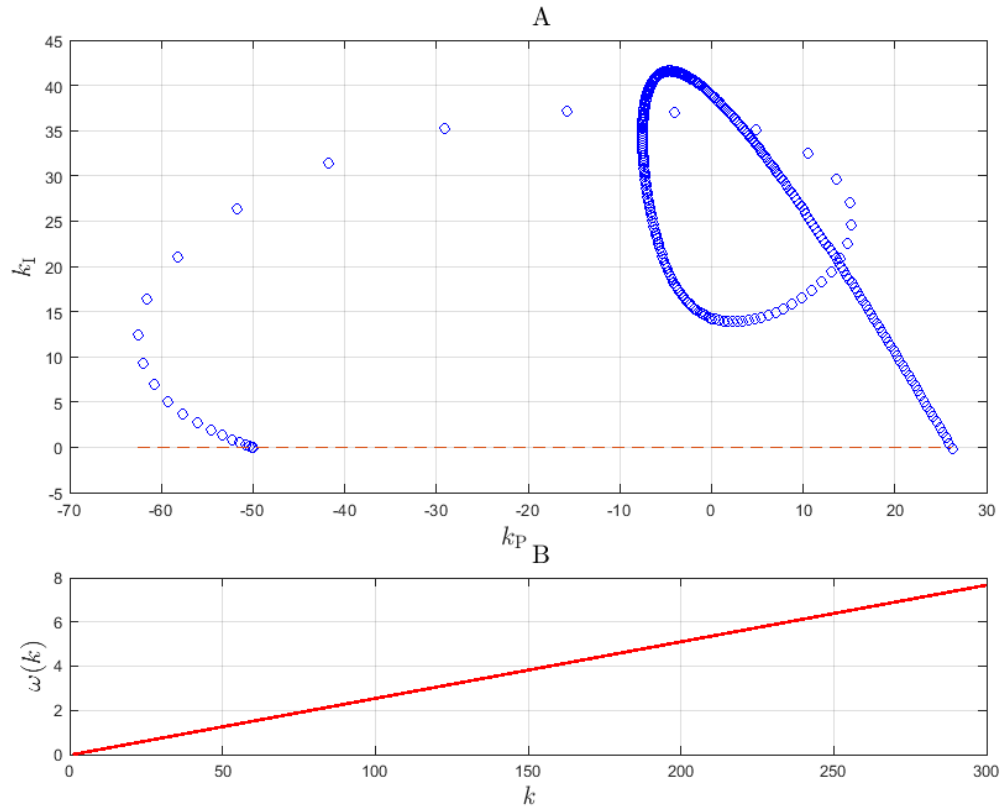
2.2.1 Výpočet hraniční křivky

Algoritmus výpočtu (resp. vykreslení) průběhu HK představuje základní část kódu, která během implementace prošla řadou změn. Celkově bylo implementováno až 10 různých způsobů výpočtu. Snahou posledních verzí algoritmu bylo zajistit co nejrovnoměrnější rozestupy bodů reprezentujících průběh HK při současné minimalizaci času potřebného k výpočtu. Jedná se o to, že pro lineární růst úhlové frekvence ω nebudou rozestupy hraniční křivky rovnoměrné, situaci ilustruje obr. 2.6. Zajištění alespoň přibližně rovnoměrných rozestupů bodů HK je však zásadní pro správné nalezení průsečíků HK, kterým se zabývá pododdíl 2.2.4.

Všechny implementované způsoby výpočtu však mají společnou základní část. Funkci je poskytnut čitatel a jmenovatel obrazového přenosu systému v polynomiálním tvaru. Nejprve je vyjádřen frekvenční přenos poskytnutého systému metodou AFRP, ten je následně metodou AEOP rozložen na sudou a lichou část (103). Dále, na základě (106), jsou pomocí MATLAB funkce `expand` získány příslušné substituce, z nichž jsou stejnojmennou funkcí na základě (108) získány parametrické vzorce, které mohou být použity k vykreslení HK v rozsahu definovaném pomocí ω_{min} a ω_{max} .

Nejjednodušší způsob výpočtu průběhu HK na základě získaných vzorců představoval algoritmus využívající *rovnoměrné rozložení hodnot* ω v intervalu $\omega = \langle \omega_{min}, \omega_{max} \rangle$ (viz

obr. 2.6 – B), kde standardně $\omega_{max} = \omega_c$. Nevýhodou však bylo *nerovnoměrné uspořádání bodů* v rovině (k_P, k_I) viz obr. 2.6 – A. Určením ω_c se zabývá pododdíl 2.2.2.



Obr. 2.6 – Vypočítaný průběh HK (A) odpovídající průběhu frekvence ω (B)

Základní postup byl dále rozšiřován tak, aby byl *zajištěn rovnoměrný rozestup*, ten je totiž stěžejní v rámci další práce s vypočítaným průběhem HK, zejména se jedná o nalezení průsečíků křivky se sebou samou.

Obecný princip zajištění alespoň přibližně rovnoměrného rozestupu bodů HK je následující:

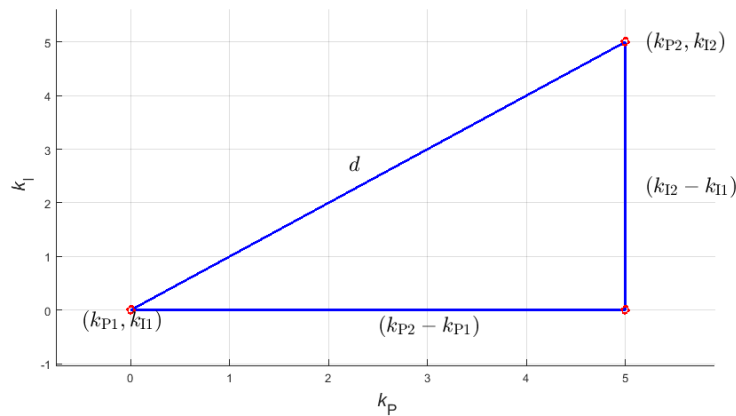
- Nejprve jsou na základě postupu uvedeném v 2.1 získány vztahy $k_P(\omega), k_I(\omega)$. Následně je v každém kroku $i = 1, 2, \dots, n$ provedeno zvýšení úhlové frekvence o drobný přírůstek $\omega = \omega + \Delta\omega$ a hodnota $k_P(\omega), k_I(\omega)$ je vypočítána. Na základě určené hodnoty k_P, k_I je následně vypočítána vhodně zvolená vzdálenostní funkce, např. euklidovská vzdálenost (viz obr. 2.7)

$$d = \sqrt{(k_{P1} - k_{P2})^2 + (k_{I1} - k_{I2})^2}, \quad (109)$$

přičemž hodnota k_P , k_I resp. ω bude zaznamenána pouze v případě, že vypočítaná vzdálenost d vyhovuje podmínce

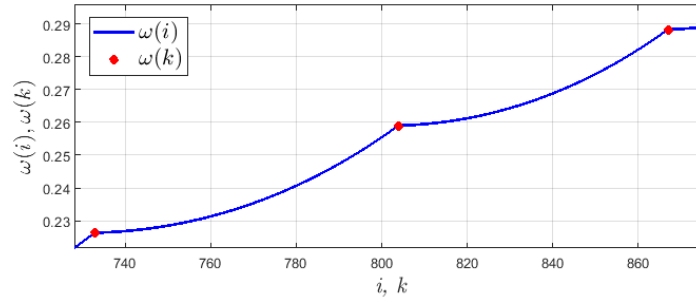
$$d > d_w. \quad (110)$$

Požadovaná vzdálenost bodů HK d_w je stanovena jako $\frac{d_{max}}{s_n}$ a je v každém kroku aktualizována v závislosti na i . Hodnota d_{max} je stanovena jako $d_{max} = d(\omega_c)$ a proměnná s_n představuje požadovaný počet bodů.



Obr. 2.7 – Stanovení následujícího bodu průběhu HK na základě splnění podmínky euklidovské vzdálenosti (110)

- Samostatným problémem je *určení rychlosti růstu* ω . Uvažování konstantní rychlosti může způsobit, že výpočet následujícího bodu bude příliš pomalý. Jako rychlostní profil se osvědčil takový, který je ovlivněn *parametrem zrychlení* a . Parametr zrychlení a je určen na základě řídicí proměnné i a je v po každém splnění podmínky (110) resetován na základní hodnotu, viz obr. 2.8. Algoritmus je rychlý a přesný, nicméně má tendenci postupně zrychlovat, to má za následek nárůst nepřesnosti pro vyšší hodnoty i .



Obr. 2.8 – Rychlostní profil $\omega(i)$, zaznamenány jsou ty hodnoty, které odpovídají podmínce (110), ty jsou označeny jako $\omega(k)$

- Další vylepšení představuje *korekční člen*. Ten spočívá ve zpřesnění základní hodnoty zrychlení a pro výpočet následujícího bodu, na základě chyby výpočtu bodu aktuálního. Uvažujme splnění podmínky (110), potom lze a pro následující krok stanovit např. jako

$$a = 1 + \frac{s_n}{w d d_{max}}. \quad (111)$$

Kde w je váhový koeficient,

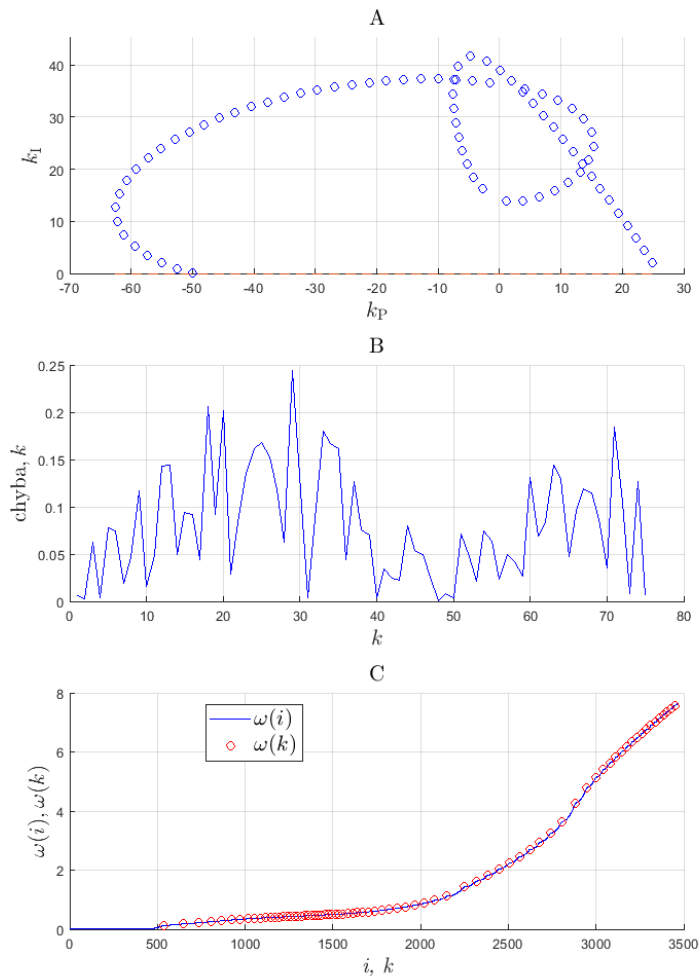
s_n – požadovaný počet bodů,

d – aktuální hodnota euklidovské vzdálenosti,

d_{max} – vypočítaná maximální hodnota euklidovské vzdálenosti.

Praktický význam přidání tohoto členu lze vysvětlit tím, že pro narůstající rychlost nárůstu parametru ω bude docházet k přesahu mezi aktuální vzdáleností d a požadovanou vzdáleností d_w . Korekce (111) způsobí, že parametr a bude tím menší, čím bude aktuální vzdálenost d větší než požadovaná d_w . Výpočet s použitím proměnné hodnoty zrychlení nemusí vést ke zvýšení přesnosti, hlavní výhodou však je urychlení výpočtu, protože hodnota zrychlení a nemusí být nastavena příliš konzervativně.

Na obr. 2.9 – A je znázorněn průběh bodů HK se zajištěním alespoň přibližných rozestupů podle postupu uvedeného výše. Obr. 2.9 – B znázorňuje chybu jako rozdíl mezi referenční d_w a skutečnou euklidovskou vzdáleností d . Obr. 2.9 – C znázorňuje odpovídající průběh argumentu ω .



Obr. 2.9 – A – průběh bodů HK se zajištěním jejich přibližných rozestupů, B – znázorňuje chybu jako rozdíl mezi referenční d_w a skutečnou euklidovskou vzdáleností d , C – znázorňuje odpovídající průběh argumentu ω

Parametry algoritmu jsou voleny jako *kompromis mezi přesností a dobou výpočtu*. Alternativně lze jako vzdálenostní funkci využít například délku křivky, u té je však potřeba vypočítat d_w na základě více než dvou hodnot, jako tomu bylo v případě euklidovské vzdálenosti. Možné je uvažovat i modifikace spočívající v zopakování výpočtu při definovaném přesahu d v aktuálním kroku nebo využít oboustranné konvergence k požadované hodnotě d .

Základní algoritmus, jak bylo uvedeno výše, spočívá ve výpočtu bodů reprezentujících průběh HK s využitím vzdálenostní funkce euklidovské vzdálenosti s rychlostním profilem a korekčním členem. V práci je využit optimalizovaný algoritmus *s postupným krokem a uvažovaným ohraničujícím obdélníkem*.

Rozšíření ve smyslu uvažovaného *ohraničujícího obdélníku* zjednodušeně znamená, že výpočet je ukončen ve chvíli, kdy průběh HK dosáhne určité hodnoty. Princip rozšířeného algoritmu spočívá v *postupném* zvyšování počáteční hodnoty ω_{max} na dvojnásobek v každém kroku k , dokud pro i (resp. k) není splněna ukončovací podmínka (116) resp. (117), nebo podmínka (118). Princip ukončení výpočtu je podrobně popsán v pododdílu 2.2.2.

2.2.2 Ukončení výpočtu průběhu hraniční křivky

Samostatným problémem je okamžik ukončení výpočtu. Jedná se v podstatě o otázku výběru konečné frekvence ω . Ta by mohla být zvolena empiricky. Vhodnější však je vypočítat ukončovací frekvenci na základě hledání kořenů rovnice

$$\Im[G(i\omega)] = 0, \quad (112)$$

tak, jak je uvedeno v článku (Tan et al., 2006). Tímto způsobem může být frekvenční rozsah rozdělen do konečného počtu intervalů.

Ukončovací frekvence ω_c představuje hodnotu, pro kterou průběh HK naposledy protne reálnou poloosu. Řešení rovnice (112) může být více. Výsledná hodnota ukončovací frekvence ω_c je stanovena jako největší reálné řešení rovnice 112. Např. pro systém

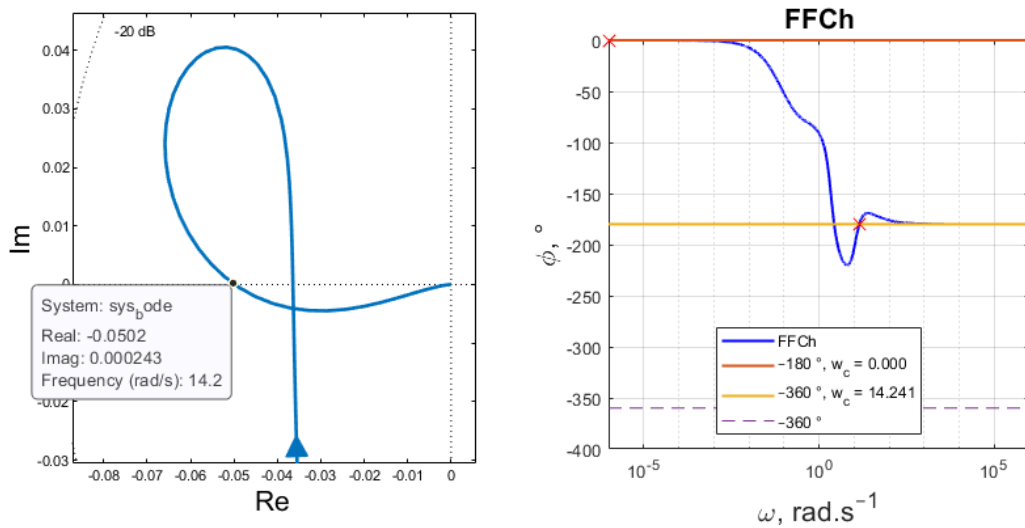
$$G(s) = \frac{s - 1}{s^2 + 0,8s - 0,2}, \quad (113)$$

je nejdříve určen frekvenční přenos a následně jeho reálná a imaginární složka

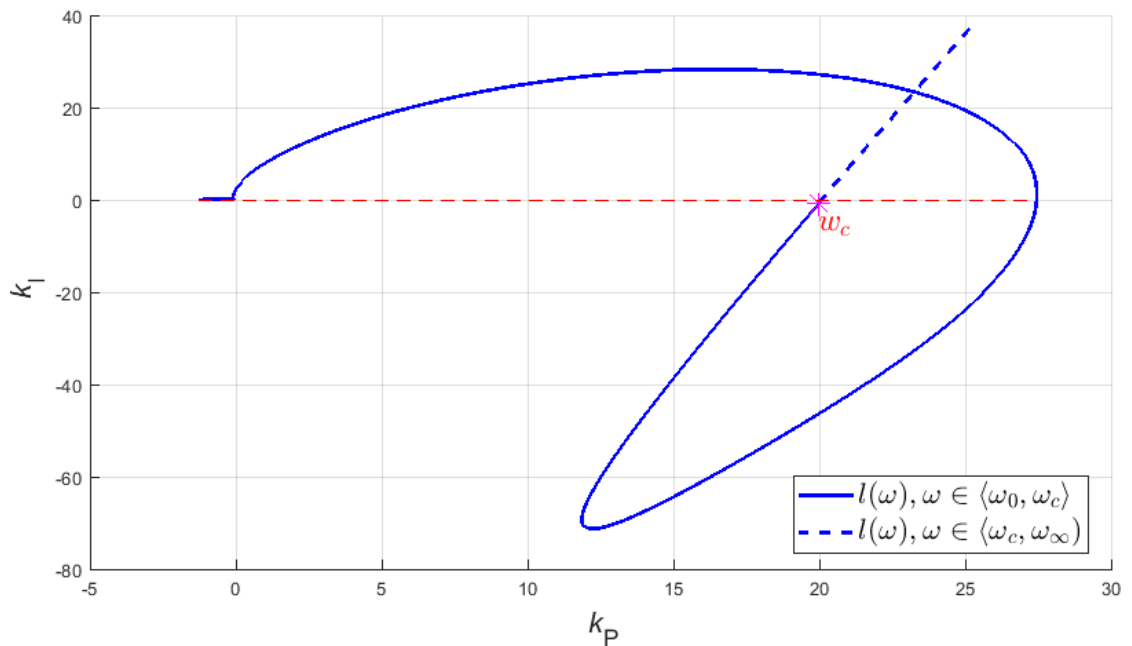
$$\begin{aligned} G(i\omega) &= \frac{i\omega - 1}{-\omega^2 + 0,8i\omega - 0,2} = \frac{i\omega - 1}{-\omega^2 - 0,2 + 0,8i\omega} \frac{-\omega^2 - 0,2 - 0,8i\omega}{-\omega^2 - 0,2 - 0,8i\omega} = \\ &= \frac{-i\omega^3 - 0,2i\omega + 0,8\omega^2 + \omega^2 + 0,2 + 0,8i\omega}{\omega^4 + 0,2\omega^2 + 0,8i\omega + 0,2\omega^2 + 0,04 + 0,16i\omega - 0,8i\omega^3 + 0,16i\omega + 0,64\omega^2} = \\ &= \frac{1,8\omega^2 + 0,2}{\omega^4 + 1,04\omega^2 + 0,04} + i \frac{-\omega^3 + 0,6\omega}{\omega^4 + 1,04\omega^2 + 0,04}. \end{aligned} \quad (114)$$

následně jsou určeny hodnoty ω , pro které platí $\Im[G(i\omega)] = 0$, řešením rovnice $-\omega^3 + 0,6\omega = 0$ jsou hodnoty $\omega_1 = 0$; $\omega_2 = 0,774$; $\omega_3 = -0,774$, přičemž ukončovací frekvence $\omega_c = 0,774$ (nejvyšší kladný kořen). To by pro uvedený systém znamenalo uvažování frekvence pro výpočet v intervalu $\omega = (0; 0,774)$.

Ukončovací frekvenci je možné určovat také na základě průsečíku FFCh s přímkou, která představuje příslušnou hladinu poklesu o $-o_r\pi$ (viz obr. 2.10), kde o_r je relativní řád systému. FFCh je využita v metodě zajišťující dostatečný pokles fáze, který je popsán v následujícím oddílu.



Obr. 2.10 – AFFCh, hledání průsečíků na základě FFCh



Obr. 2.11 – Ukázka nevhodného ukončení vypočítanou (nalezenou) $\omega_c = 14,24$ (značena fialovou hvězdou) – průběh HK by dále pokračoval ve vymezování oblastí

Později se však ukázalo, že ukončení výpočtu průběhu HK v bodě ω_c tak, jak je uvedeno v (Tan et al., 2006), nestačí. Uvažujme následující příklad

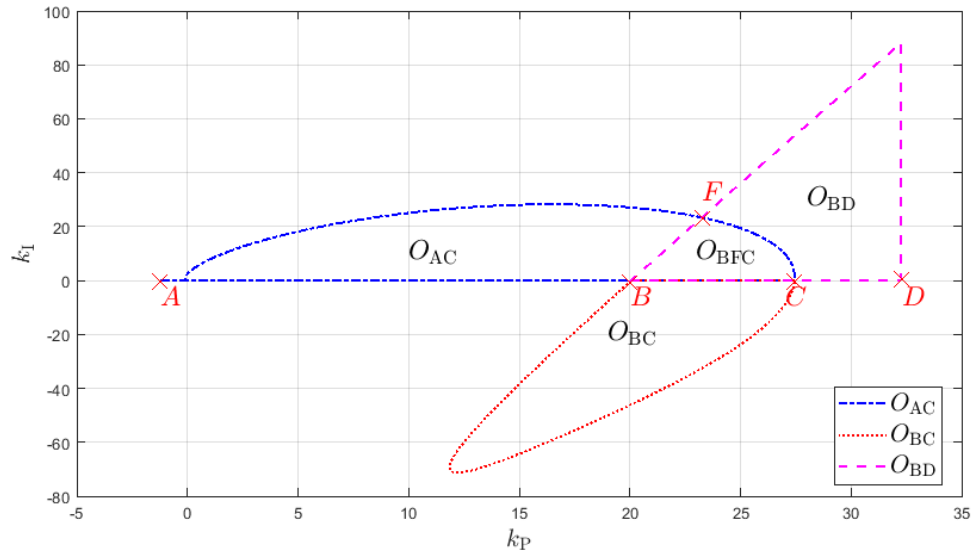
$$G(s) = \frac{10s^5 - 50s^4 + 4s^3 + 10s^2 - 500s + 40}{s^7 + 2s^6 + 3s^5 + 1410s^4 + 40s^3 + 50s^2 + 4s + 50}. \quad (115)$$

Ukončovací frekvence je pro tento systém analyticky určena jako $\omega_c = 14, 24$ a průběh pro rozsah $\omega \in (0; 14, 24)$ je na obr. 2.11. Nicméně, kdyby křivka nebyla ukončena v bodě $\omega_c = 14, 24$, tak by svým průběhem rozdělila rovinu (k_P, k_I) na další podoblast.

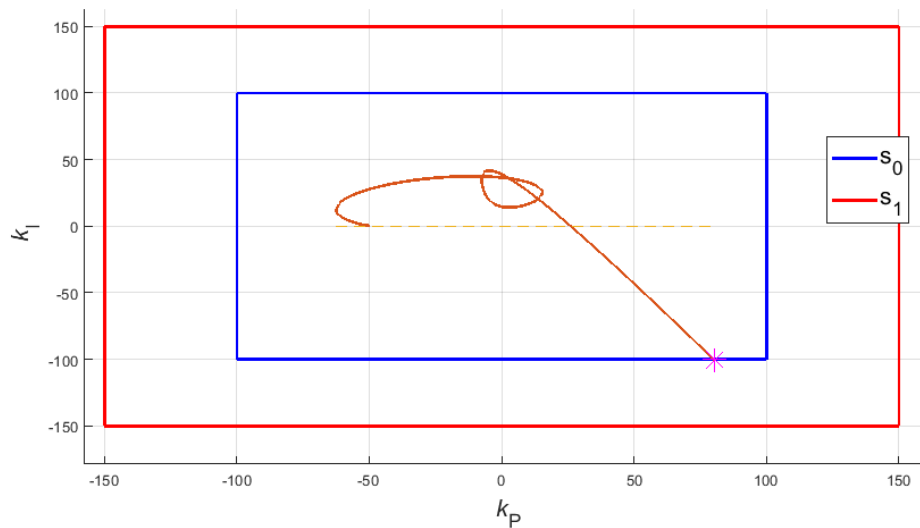
Z toho vyplývá, že *není vhodné ukončovat výpočet nalezenou ukončovací frekvencí ω_c* , protože pak by se mohlo stát, že některá oblast nebude nalezena. To však platí v případě, že se HK přibližuje k ose k_I zdola. V případě, že se HK přibližuje k ose $k_I = 0$ shora, tak je možné výpočet ukončit nalezenou ω_c , protože platí, že *nemá smysl pracovat s oblastmi*, kde $k_I \leq 0$, za předpokladu, že systém má *kladné zesílení* ve smyslu poměru absolutních členů pro $\omega = 0$, přičemž kladné zesílení zajišťuje funkce normalizace přenosu, jejíž princip lze nalézt v pododdílu 2.2.3.

Jakmile je zajištěno kladné zesílení systému, tak musí mít integrační složka regulátoru *kladnou hodnotu*, v opačném případě by došlo k realizaci kladné zpětné vazby. URO může být stabilní i v případě kladné zpětné vazby, nicméně nedojde k ustálení trvalé regulační odchylky a v konečném důsledku nebude možné vypočítat integrační kritérium regulační plochy.

Dále autoři článku (Tan et al., 2006) zmiňují testování stability *mezi sousedními průsečíky* s $k_I = 0$, tedy mezi dvojicemi kořenů nalezených pomocí (112). Hledat oblasti mezi sousedními průsečíky s $k_I = 0$ však není výhodné i v případě, že by v místě posledního průsečíku směřovala HK směrem dolů. Uvažujme situaci na obr. 2.12 – mezi body A, B, C, D, které odpovídají kořenům rovnice (112), by byla nalezena oblast O_{AC} (mezi AC), O_{BD} (mezi BD) a O_{BC} (mezi BC). Oblast O_{BFC} (mezi BCF) by však nalezena nebyla. Takto vymezené oblasti navíc nejsou reprezentovány korektně, protože oblast O_{AC} a O_{BD} mají společný průnik. Toto představuje významný poznatek z hlediska implementace, který v důsledku vedl na nutnost komplexnějšího řešení problému.



Obr. 2.12 – Oblast O_{AC} , O_{BC} a O_{BD} , které se nachází mezi dvojicemi průsečíků A, B, C, D s osou $k_I = 0$.



Obr. 2.13 – Ohraničující obdélník pro parametr s_0 (podmínka (116), resp. (117)) a parametr s_1 (podmínka (118))

Problematické situace uvedené výše lze řešit následující volbou ukončení průběhu výpočtu. Krok i se zvyšuje, dokud není detekováno překročení ohraničujícího obdélníku (viz obr. 2.13)

$$|k_P(i)| \geq s_0 \vee |k_I(i)| \geq s_0, \quad (116)$$

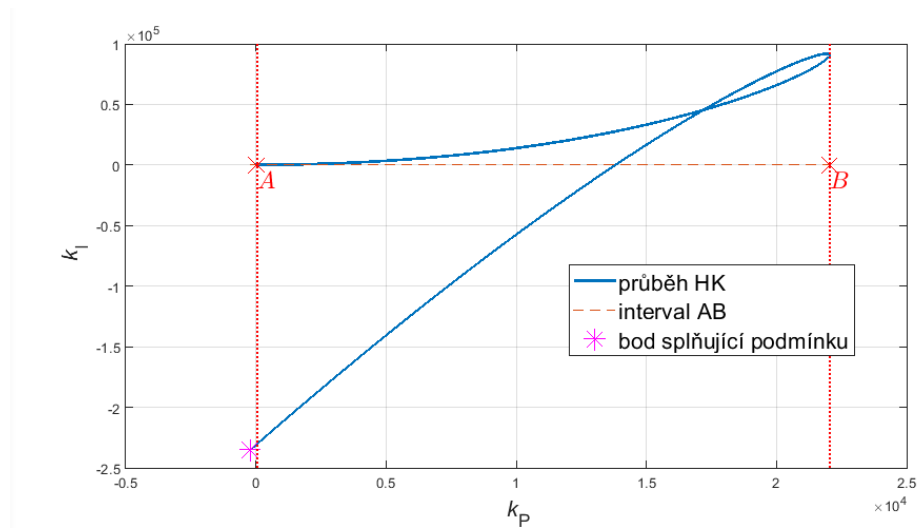
a současně není splněna podmínka

$$k_P(i) \notin (A, B), \quad (117)$$

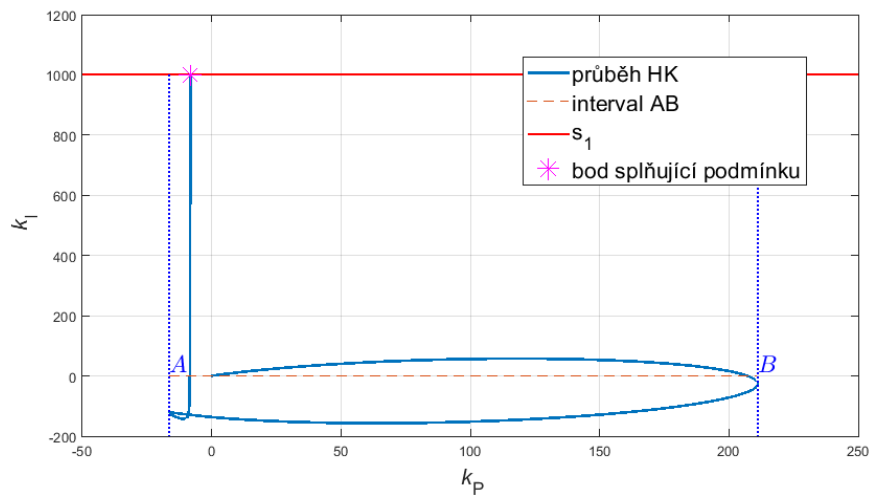
kde (A, B) je ohraničující interval v ose k_P , viz např. obr. 2.14. Algoritmus bude ukončen i v případě, že bude splněna podmínka

$$|k_I| \geq s_1, \quad (118)$$

kde s_1 představuje mez ukončení bez ohledu na podmínku (117), viz např. obr. 2.15 a typicky $s_1 > s_0$.

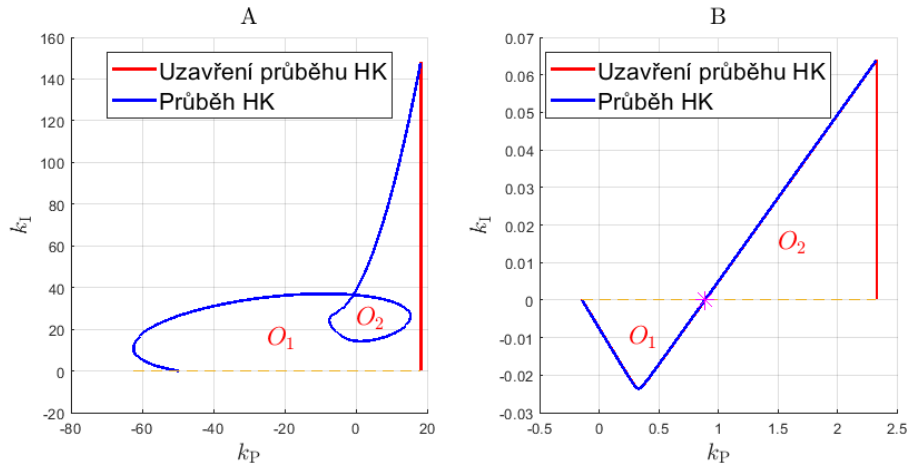


Obr. 2.14 – Koncový bod HK vyhovující podmínce (116), resp. (117) ($s_0 = 10^3$ není v obrázku vyznačena)



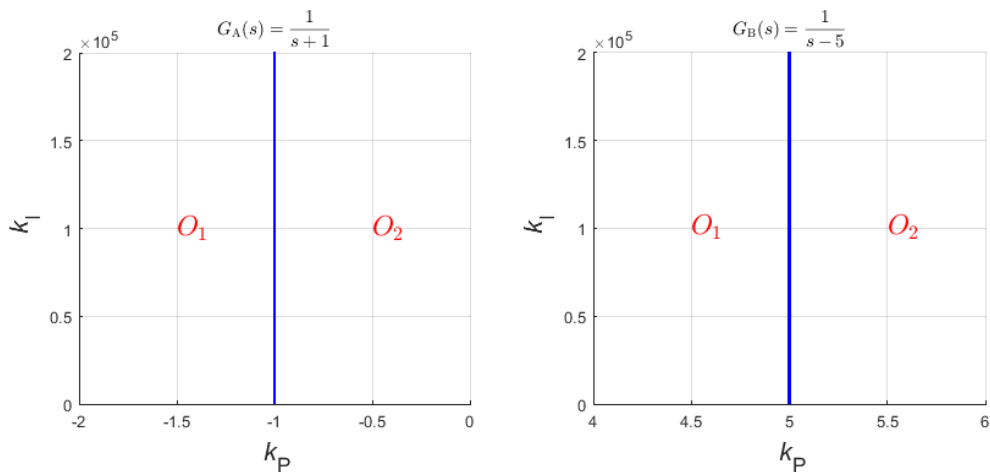
Obr. 2.15 – Koncový bod HK vyhovující podmínce (118), pro $s_1 = 10^3$

V souvislosti s ukončením výpočtu tímto způsobem je potřeba průběh HK křivky *uzavřít*, tuto situaci zachycuje obr. 2.16 – A, B. Obrázek zároveň prezentuje výhodu tohoto způsobu ukončení výpočtu – např. oblast O_2 na obr. 2.16 – A je vymezená průběhem HK, pro který však nebyla nalezena ukončovací frekvence ω_c . Stabilní oblast O_2 na obr. 2.16 – B by zase nebyla vůbec nalezena v případě, že by výpočet průběhu HK byl ukončen nalezenou ukončovací frekvencí ω_c .



Obr. 2.16 – A – ukončovací frekvenci nelze na úrovni výpočtu nalézt nebo je příliš vysoká, B – stabilní oblast vymezi až průběh hraniční křivky za ω_c (fialová hvězda)

Metoda HK není vhodná pro systémy 1. řádu (resp. systémy s pouze jedním pólem). V takovém případě bude totiž hraniční křivka neustále růst jedním směrem (podle znaménka zesílení) a svírat s osou $k_I = 0$ pravý úhel, viz obr. 2.17. Situaci lze řešit přidáním malé časové konstanty do jmenovatele přenosu, viz pododдіl 2.2.3.



Obr. 2.17 – Ilustrace průběhu HK systémů s pouze jedním pólem

2.2.3 Úprava přenosu uvažovaného systému

Základní operací této metody je *normalizace* přenosu ve smyslu zajištění kladného zesílení (16). Normalizace na kladné zesílení spočívá ve vynásobení čitatele přenosu systému znaménkem poměru absolutních členů p . Tedy

$$G_n(s) = \frac{Np}{D}, \quad (119)$$

kde N je čítec obrazového přenosu systému,

D – jmenovatel obrazového přenosu systému,

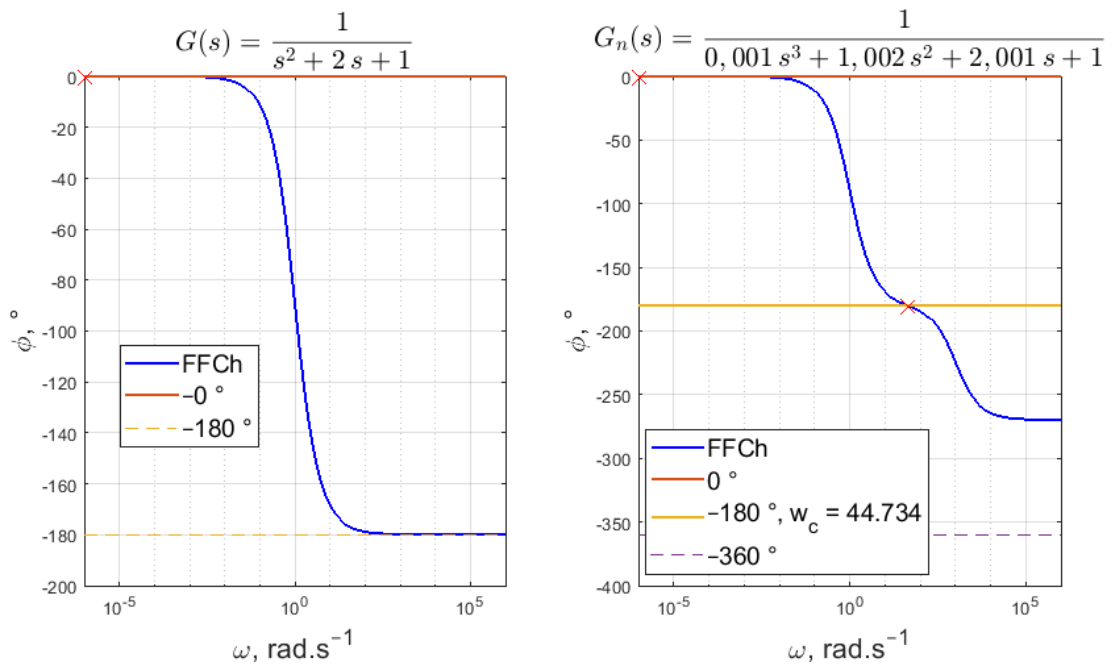
přičemž

$$p = \operatorname{sgn} \left(\frac{b_0}{a_0} \right). \quad (120)$$

Dále jsou uvažovány obrazové přenosy systémů upravené tímto způsobem. Z toho vyplývá, že koeficientem p je potřeba vynásobit i výsledné nalezené nastavitelné parametry PI(D) regulátoru.

Dále je potřeba řešit situace, kdy HK pro nalezenou ω_c původního systému směřuje (za posledním průsečíkem s $k_I = 0$) ke *kladným* hodnotám k_I (viz např. obr. 2.11). Metody nalezení ukončovací kritické frekvence ω_c lze nalézt v pododdílu 2.2.2. Tento problém lze řešit tak, že se obrazový přenos původního systému $G(s)$ *rozšíří dynamickým členem* s vhodnou časovou konstantou, která je zároveň zanedbatelná oproti časovým konstantám původního systému $G(s)$. Systém je takto rozšiřován do té doby, dokud není zajištěno, že se v okolí ω_c rozšířeného systému průběh hraniční křivky $l(\omega_c)$ blíží k ose $k_I = 0$ shora.

Úprava přenosu tímto způsobem řeší i situace, kdy ω_c není nalezena, protože systém má příliš nízký o_r (kde o_r je relativní řád systému) a URO tedy *nelze* změnou parametrů regulátoru *přivést na kmitavou mez stability*. Rozšíření má tedy zajistit, že $o_r \geq 3$. To má za následek nalezení průsečíku FFCh s poklesem fáze o -180° , viz obr. 2.18.



Obr. 2.18 – FFCh původního (nerozšířeného) a FFCh rozšířeného systému

Obecnou funkci algoritmu lze popsat následovně:

1. Normalizuj přenos ve smyslu zajištění kladného zesílení vynásobením čitatele přenosu znaménkem poměru absolutních členů.
2. Pokud je funkce volána i s požadavkem na rozšíření přenosu $scale = 1$, tak:
 - a) Pomocí MATLAB funkce `damp` urči pole vlastních frekvencí ω_n systému a následně pole časových konstant systému $\tau = \frac{1}{\omega_n}$, urči také $T_{min} = \frac{\min(\tau)}{10^3}$.
 - b) Dokud je $\phi_{min} > \phi_{init} - 180$ (kde ϕ_{init} značí počáteční fázový posuv systému a ϕ_{min} značí minimální fázový posuv systému), tak

$$G_n(s) = \frac{N(s)}{D(s)} \frac{1}{(T_{min}s + 1)^k}, \quad (121)$$

kde k se postupně zvyšuje.

3. Dokud platí, že se HK přibližuje k ose $k_I = 0$ zezdola, tak $k = k + 1$ a opakuj (121).

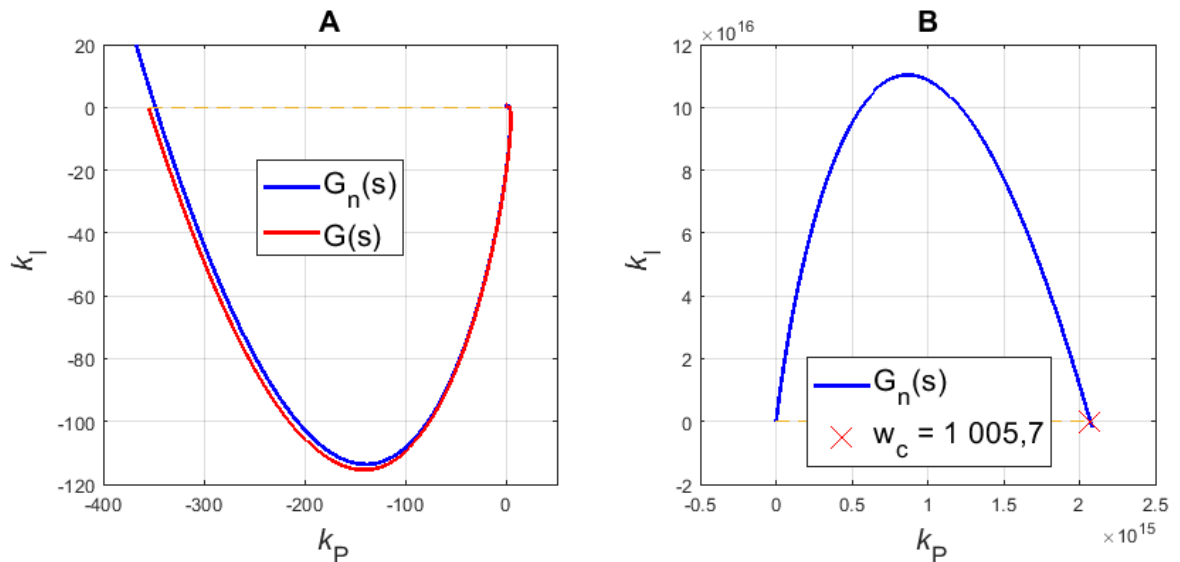
Příklad – na obr. 2.19 je vidět, že průběh HK odpovídající původnímu systému

$$G(s) = \frac{1}{1,0s^5 + 5s^4 + 10s^3 + 10s^2 + 5s + 1} \quad (122)$$

by směřoval nahoru, když by nebyl ukončen vypočítanou ω_c . Po provedení rozšíření je však rozšířen na přenos

$$G_n(s) = \frac{1}{9,9e^{-7}s^7 + 0,002s^6 + 1,01s^5 + 5,02s^4 + 10,02s^3 + 10,01s^2 + 5,002s + 1},$$

pro který je stanovena ukončovací kritická frekvence $\omega_c = 1005,7$ a odpovídající průběh tohoto rozšířeného přenosu je vykreslen pro porovnání. V rámci zvýšení míry shody s HK původního systému by bylo možné použít menší hodnotu parametru T_{min} .



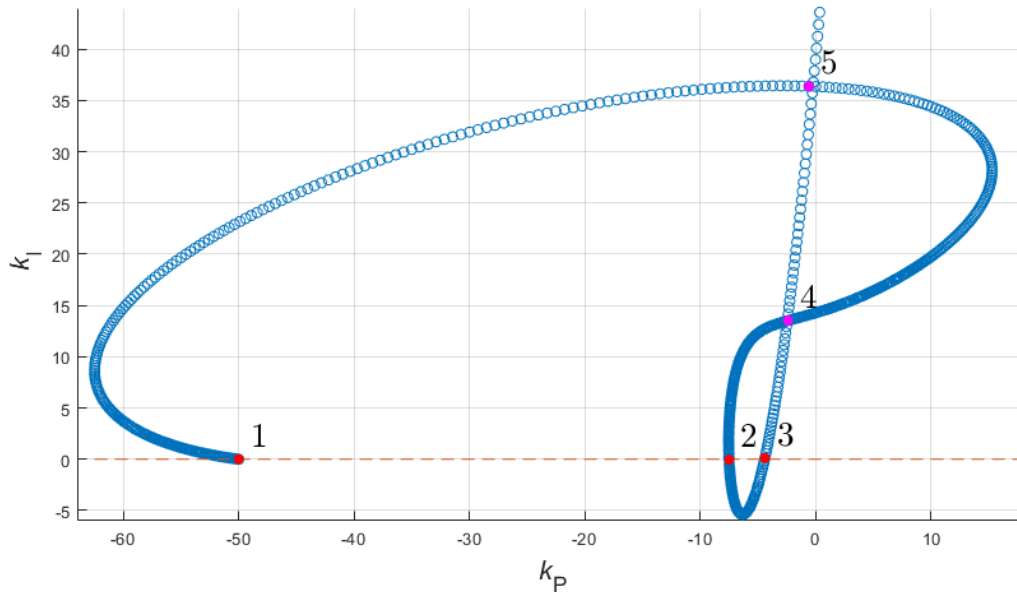
Obr. 2.19 – Průběh HK původního (122) a rozšířeného (123) systému, A – nekorektní ukončení průběhu původního systému a srovnání míry přesnosti s průběhem HK rozšířeného systému, B – korektní ukončení průběhu HK rozšířeného systému

Důvodem rozšíření je to, že pokračující HK může rozdělit již nalezenou podoblast na 2 dílčí podoblasti. Za *nevýhodu* rozšíření přenosu původního systému o dodatečné členy se zanedbatelnou dynamikou lze považovat nesoulad průběhu HK původního a rozšířeného systému a také skutečnost, že poslední oblast, která neodpovídá původnímu systému již nemusí být středem zájmu i v případě, že odpovídá stabilizujícím parametrům regulátoru. Nevýhodou také je, že poslední oblast rozšířeného systému může být příliš rozsáhlá, např. průběh HK rozšířeného systému na obr. 2.19 – B končí až v hodnotě $k_p \approx 2,1 \cdot 10^{15}$.

Na základě výše zmíněných poznatků je výpočet ukončován nikoli na základě vypočítané hodnoty ω_c , nýbrž v momentě, kdy HK původního nerozšířeného systému dosáhne saturační meze, tzn. splnění podmínky (116), resp. (117), nebo podmínky (118). Vypočítanou hodnotu ω_c je však možné využít v rámci původní metody výpočtu průběhu HK, která je součástí přiloženého kódu a která zajistí výpočet HK v rozsahu úhlových frekvencí w_{min} a ω_c .

2.2.4 Získání významných průsečíků průběhu hraniční křivky

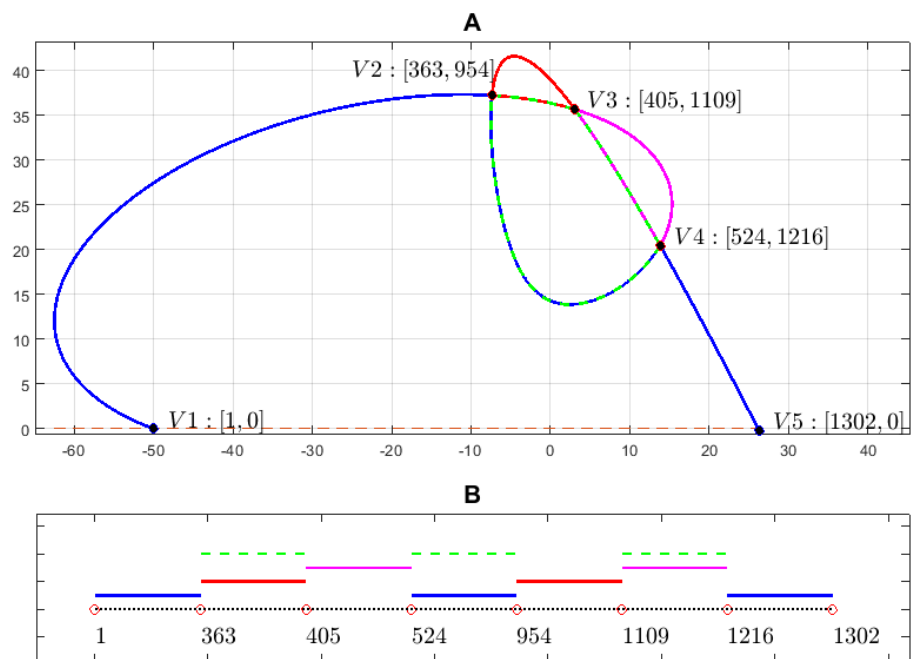
Úloha hledání průsečíku HK je stěžejní, průsečíky vymezují segmenty hraniční křivky, které definují oblasti v rovině. Jedná se jednak o úlohu nalezení průsečíků HK se sebou samou a také nalezení průsečíků s osou $k_I = 0$, resp. jejich indexů, při současném zachování jejich pořadí, tak, jak ho vymezení průběh HK $l(\omega)$ pro narůstající ω , viz obr. 2.20.



Obr. 2.20 – Pořadí bodů reprezentujících významné průsečíky

V případě, že se jedná o průsečík křivky se sebou samou, tak je výstupem dvojice indexů v poli, které reprezentuje průběh HK. Tato dvojice indexů odpovídá stejné hodnotě nalezené souřadnice průsečíku v rovině (k_P, k_I) , taková souřadnice bude nazývána standardním uzlem (SU, viz $V2, V3, V4$ na obr. 2.21 – A). Kromě standardních uzlů je třeba rozlišovat i tzv. koncové uzly (KU), ty odpovídají průsečíku HK s osou $k_I = 0$, a dané souřadnici odpovídá pouze jeden index (viz $V1, V5$ na obr. 2.21 – A).

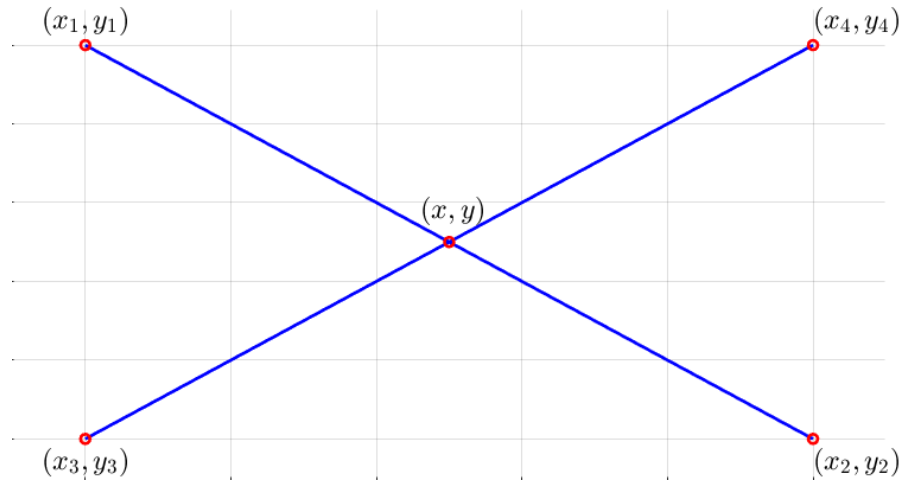
Nalezené průsečíky (resp. jejich indexy) představují základní údaj, na základě kterého lze následně jednotlivé oblasti sestavit, to demonstruje obr. 2.21. Jednotlivé oblasti na obr. 2.21 – A lze sestavit uvažováním úseků pole reprezentujícího průběh HK, které se nacházejí mezi jednotlivými indexy průsečíků, viz obr. 2.21 – B.



Obr. 2.21 – A – zachycuje průběh HK a jednotlivé vymezené oblasti, B – ilustruje rozsahy indexů průsečíků reprezentujících jednotlivé oblasti v poli popisujícím průběh HK

Nalezení souřadnic průsečíků

Výpočet HK a úloha nalezení průsečíků HK byly odděleny, je tedy prováděno hledání průsečíku nad dostupnými daty, které popisují průběh HK. Souřadnice průsečíků v rovině (k_P, k_I) jsou vypočítávány jako průsečíky lineárních segmentů mezi jednotlivými body (viz obr. 2.22), které popisují průběh HK. Alternativně by, v rámci zvýšení přesnosti souřadnic průsečíků, bylo možné hledat průsečíky úseků polynomem proložených bodů, či k tomuto účelu využít metody půlení intervalu, to však nemá praktický význam.



Obr. 2.22 – Znáznornění průsečíku 2 lineárních segmentů

Úloha hledání průsečíku představuje časově nejnáročnější (narůstá s druhou mocninou vyšetřovaných dvojic lineárních segmentů, jejichž počet navíc naroste případným následným proložením) a z hlediska chyby i kritickou část kódu. Vlastní implementace algoritmů pro nalezení průsečíků nebyly z hlediska výpočetní náročnosti příliš efektivní. Z tohoto důvodu byla použita z internetu převzatá funkce `intersections` (Schwarz, 2021). Ta funguje následovně – mějme 2 úseky křivky L_1 a L_2 a jejich koncové body (KB, viz obr. 2.22)

$$\begin{aligned} L_1 &: (x_1, y_1), (x_2, y_2), \\ L_2 &: (x_3, y_3), (x_4, y_4). \end{aligned} \tag{124}$$

Potom lze zapsat 4 rovnice se 4 proměnnými a ty vyřešit. Tyto 4 proměnné jsou t_1 , t_2 , x_0 a y_0 , kde (x_0, y_0) je průsečík mezi L_1 a L_2 , parametr t_1 představuje vzdálenost od počátku segmentu L_1 do bodu průsečíku relativně k délce L_1 a t_2 je vzdálenost od počátku segmentu L_2 do bodu průsečíku relativně ke vzdálenosti L_2 . Tyto 4 rovnice tedy jsou

$$\begin{aligned} (x_2 - x_1)t_1 &= x - x_1, \\ (x_4 - x_3)t_2 &= x - x_3, \\ (y_2 - y_1)t_1 &= y - y_1, \\ (y_4 - y_3)t_2 &= y - y_3. \end{aligned} \tag{125}$$

Přeuspořádáním a zápisem v maticové formě

$$\underbrace{\begin{bmatrix} x_2 - x_1 & 0 & -1 & 0 \\ 0 & x_4 - x_3 & -1 & 0 \\ y_2 - y_1 & 0 & 0 & -1 \\ 0 & y_4 - y_3 & 0 & -1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} t_1 \\ t_2 \\ x \\ y \end{bmatrix}}_T = \underbrace{\begin{bmatrix} -x_1 \\ -x_3 \\ -y_1 \\ -y_3 \end{bmatrix}}_B. \quad (126)$$

Následně lze vyřešit

$$T = A^{-1}B, \quad (127)$$

jakmile je získáno řešení, tak zbývá vyšetřit t_1 a t_2 ke zjištění, zda-li se L_1 a L_2 protínají.

Pokud platí

$$\begin{aligned} 0 &\leq t_1 < 1, \\ 0 &\leq t_2 < 1, \end{aligned} \quad (128)$$

tak se dané úseky protínají a souřadnice průsečíku (x, y) je zahrnuta do výstupu.

V principu je potřeba provést tento výpočet na každém lineárním segmentu křivky – dvojici hodnot ve vstupních datech. To může být poměrně vysoký počet dvojic, takže je implementována i část kódu, která předběžným výpočtem vyloučí určité segmenty, které se nemohou protínat. Tato část kódu spočívá ve vyšetření existence společného průniku obdélníků (jejichž rozměr je dán rozdílem (x, y) hodnot, se stranami paralelně k osám) pro každou dvojici reprezentující lineární segment křivky a zjištění, zda-li se dané obdélníky překrývají. Typicky tato část kódu vyřadí většinu dvojic bodů. Pokud se dané obdélníky překrývají, tak je potřeba v rámci zjištění, zda-li se úseky protínají, vypočítat t_1 a t_2 dle (127).

Na základě poznatků analytické geometrie lze využít *alternativní formulace* vztahů (124) ÷ (128). Výchozí je předpoklad, že průsečík mezi přímkami v rovině existuje vždy, s výjimkou případu, kdy jsou rovnoběžné. Uvažujeme-li úsečky, tak se jedná o vyšetření, zda-li je průsečík v daném rozsahu. Existence průsečíku mezi 2 segmenty lze vyšetřit také pomocí výpočtu

$$d_1 = \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix} \Bigg| \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_4 \\ y_1 & y_2 & y_4 \end{vmatrix}, \quad (129)$$

a

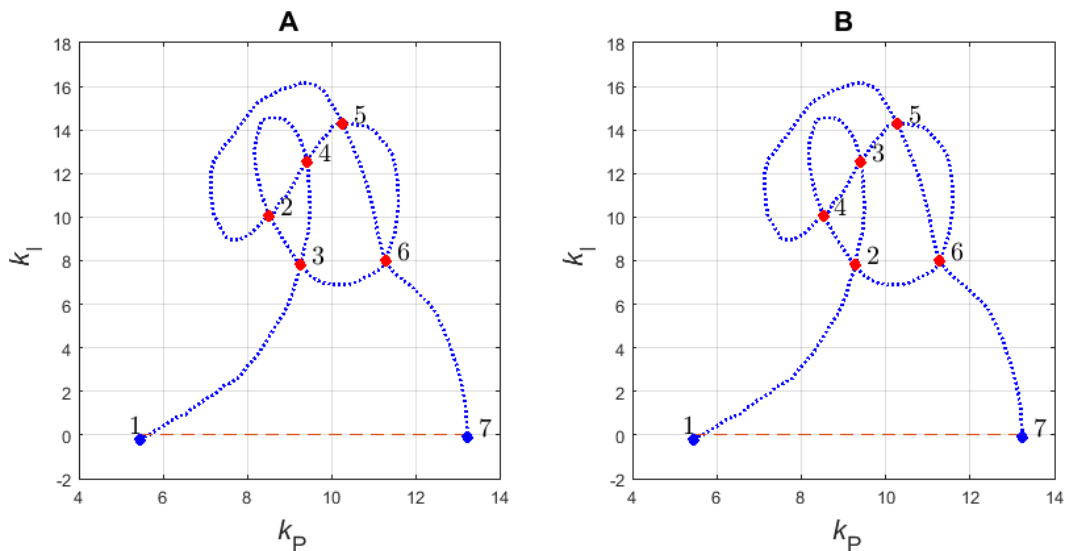
$$d_2 = \left| \begin{array}{ccc|ccc} 1 & 1 & 1 & 1 & 1 & 1 \\ x_1 & x_3 & x_4 & x_2 & x_3 & x_4 \\ y_1 & y_3 & y_4 & y_2 & y_3 & y_4 \end{array} \right|. \quad (130)$$

V případě, že mezi danými segmenty existuje průsečík, musí platit

$$d_1 \leq 0 \wedge d_2 \leq 0. \quad (131)$$

Nalezení správného pořadí indexů průsečíků v poli popisujícím průběh HK

Nevýhodou metody `intersections` je, že současně nevrací indexy v poli, které odpovídají nalezeným (x, y) souřadnicím průsečíků, a to, že získané souřadnice (průsečíky) nejsou seřazeny (obr. 2.23 – A) tak, jak by byly nalezeny postupným prohledáváním pole reprezentujícího průběh HK, viz obr. 2.23 – B. Tyto 2 požadavky jsou nezbytné a proto byly v rámci jejich zajištění vytvořeny optimalizované algoritmy, které představují nadstavbu této metody.



Obr. 2.23 – Nesprávné (A) a správné řazení průsečíků (B)

Nadstavba funkce `intersections` spočívá v nalezení indexů, které odpovídají všem významným průsečíkům, a dále také v zajištění jejich správného pořadí. Vstupem algoritmu je pole xy popisující průběh HK. Výstupem je standardizovaná matice `out`, která obsahuje

indexy všech významných průsečíků a dále obsahuje informaci o tom, zda-li se jedná o průsečík křivky se sebou samou, či o průsečík křivky s osou $k_I = 0$.

Obecný princip algoritmu nalezení správného pořadí indexů všech významných průsečíků je následující:

1. Nejprve jsou pomocí funkce `intersections` nalezeny souřadnice průsečíků křivky se sebou samou xy_p . Následně je vypočítána délka křivky d mezi jednotlivými body xy , které reprezentují průběh HK. Dále je pro všechny $i = 1, 2, \dots, n$, kde n je počet bodů HK a následně pro všechna $j = 1, 2, \dots, m$, kde m je celkový počet vypočítaných průsečíků, vypočítána vzdálenost D mezi $xy(i)$ a $xy_p(j)$. Indexy průsečíku (i, j) jsou zaznamenány, je-li vypočítaná vzdálenost $D(i, j)$ menší než $d(i)$. Tímto jsou získány indexy P v poli xy , které odpovídají vypočítaným souřadnicím průsečíků. Pro příklad na obr. 2.24 byla získána matice

$$P = \begin{pmatrix} 25 & 33 & 47 & 50 & 60 & 73 & 78 & 83 & 112 & 122 \\ 2 & 3 & 1 & 2 & 5 & 4 & 3 & 1 & 4 & 5 \end{pmatrix}^T. \quad (132)$$

2. Následně je potřeba uspořádat takto získané průsečíky do dvojic indexů, které odpovídají danému vypočítanému průsečíku, v pořadí, v jakém je průběh HK vymezil. Matice SU je vytvořena z hodnot prvního sloupce matice P , pro stejné hodnoty ve druhém řádku matice P . Matice SU bude mít pro (132) tvar

$$SU = \begin{pmatrix} 25 & 33 & 47 & 60 & 73 \\ 50 & 78 & 83 & 122 & 112 \end{pmatrix}^T. \quad (133)$$

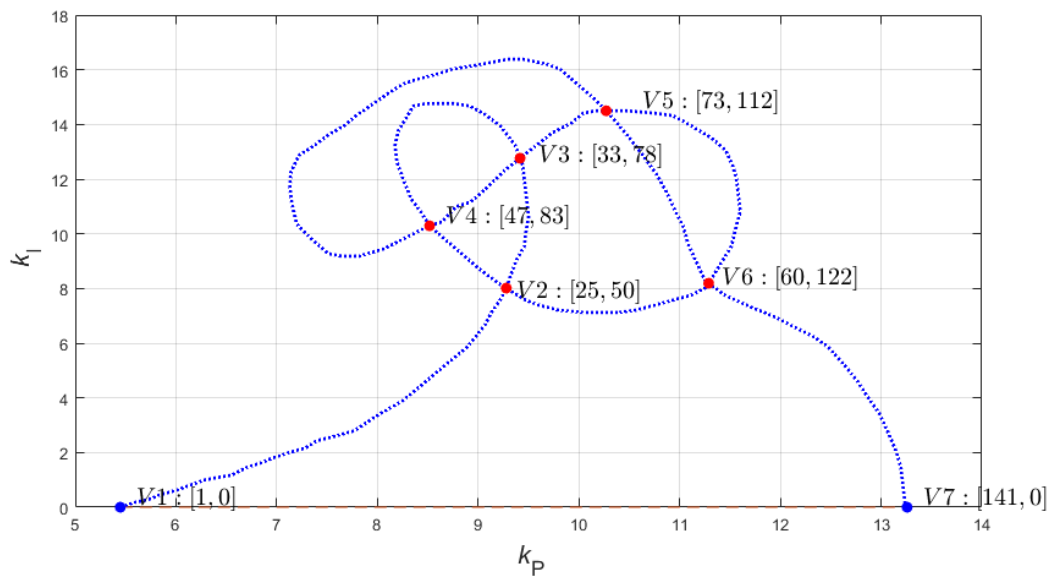
3. Dále je pro všechna $i = 2, 3, \dots, n$ nalezen index průsečíku s osou $k_I = 0$. Dojde-li mezi $y(i)$ a $y(i-1)$ ke znaménkové změně, tak je jako index průsečíku zaznamenána hodnota $(i, 0)$. Matice KU všech takto nalezených průsečíků pro příklad na obr. 2.24 má tvar

$$KU = \begin{pmatrix} 1 & 141 \\ 0 & 0 \end{pmatrix}^T. \quad (134)$$

4. Nakonec je potřeba vytvořit matici *out*, která vznikne z matic SU a KU a to následovně. Jednotlivé řádky matice SU jsou společně s řádky matice KU zahrnuty do matice *out* tak, že nejmenší, resp. největší hodnota řádku SU není menší, resp.

větší než nejmenší, resp. největší hodnota na řádce matice KU . Matice out pro (133), (134) má tvar

$$out = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 25 & 33 & 47 & 73 & 60 & 141 \\ 0 & 50 & 78 & 83 & 112 & 122 & 0 \end{pmatrix}^T. \quad (135)$$



Obr. 2.24 – Průsečíky popsané prostřednictvím nalezených indexů a zajištění sekvenčního seřazení

Algoritmus 1. Zajišťující nalezení správného pořadí indexů všech významných průsečíku je následující:

1. Na dvojici xy popisujících průběh HK nalezni funkci `intersections` (Schwarz, 2021) souřadnice průsečíků křivky se sebou samou xy_0 . Stanov Lx jako délku pole xy a Lx_0 jako délku pole xy_0 .
2. Pro všechna $i = 2, \dots, Lx$ vypočítej vzdálenost `arclen(i)` (pomocí funkce `arclength` (D'Errico, 2021)) mezi bodem i a bodem $i - 1$.
 - a) Pro $j = 1, \dots, Lx_0$ vypočítej d jako euklidovskou vzdálenost mezi souřadnicí průsečíku j a souřadnicí bodu i , pokud $d < arclen(i)$, potom zaznamenej $P = [i, j]$. Odstraň duplicitní řádek P v případě, že vznikne. Pro vnější cyklus nastav proměnnou, která neumožní vstup do těla tohoto cyklu po 1 iteraci řídicí proměnné i .
3. Pro $i = 1 : Pr$, kde Pr představuje počet řádků matice P .
 - a) urči $v = P(i, 2)$,

- b) nalezni všechny řádky ve 2. sloupci matice P , které obsahují v , a ty označ jako pole s v případě, že se všechny prvky druhého sloupce matice rovnají v ,
 - c) vytvoř řádek matice SU (SU – standardní uzel) obsahující hodnoty z 1. sloupce s -tého řádku matice P .
4. Proveď deduplikaci řádků matice SU .
 5. Pro $i = 2, \dots, Lx$.
 - a) zjistí, zdali došlo ke změně znaménka mezi bodem i a bodem $i - 1$, a pokud ano, tak $KU = [i, 0]$ (KU – koncový uzel), znemožni vstupu do těla podmínky po dobu jedné iterace řídicí proměnné i .
 6. Vertikálně spoj SU s KU jako A , zleva přidej sloupec označující sestupné pořadí do A , na každém řádku 2. a 3. sloupce A nalezni nejvyšší hodnotu a toto pole hodnot seřaď vzestupně a indexy odpovídající řazení označ jako o , přeorganizuj řádky A podle pole o , aktualizuj první sloupec matice A na původně nastavenou hodnotu, označ $out = A$.

2.2.5 Využití teorie grafů v souvislosti s metodou hraniční křivky

Neuspořádaná dvojice prvků a a b zapisovaná jako (a, b) je množina, která obsahuje jen a pouze prvky a a b . Pořadí, ve kterém jsou prvky uvedeny, nehraje roli, Množina všech neuspořádaných dvojic prvků z množiny V se značí $V \otimes V$. Například pro množinu $V = \{1, 2, 3\}$ je množina všech neuspořádaných dvojic

$$V \otimes V = \{(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)\}, \quad (136)$$

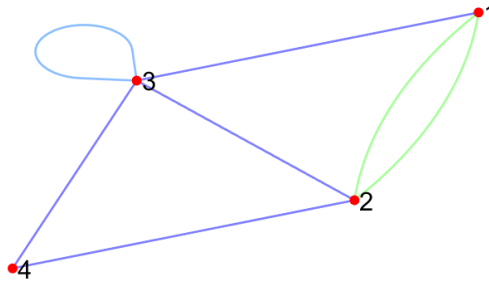
přičemž přirozeně platí $(1, 2) = (2, 1)$.

Neorientovaný grafem rozumíme uspořádanou trojici $G = (V, X, p)$, kde V je množina vrcholů (resp. uzlů), X je množina hran a p je zobrazení (Rak, 2017)

$$p : X \longrightarrow V \otimes V. \quad (137)$$

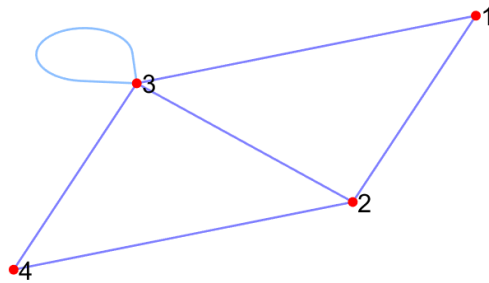
Zobrazení p se nazývá incidence, ta přiřazuje každé hraně neorientovaného grafu G dvojici vrcholů. Je-li incidence hrany $h \in X : p(h) = (u, v)$, hovoříme, že hrana h inciduje s vrcholy u a v . Jsou-li vrcholy u a v spojeny hranou, nazýváme je vrcholy přilehlými (Rak, 2017).

Nejobecnějším typem grafu je *multigraf*, ten připouští existenci více různých hran spojujících stejné vrcholy, ty se nazývají *rovnoběžné hrany* a připouští i existenci hran, které spojují vrchol v s vrcholem v , ty se nazývají *smyčka*.



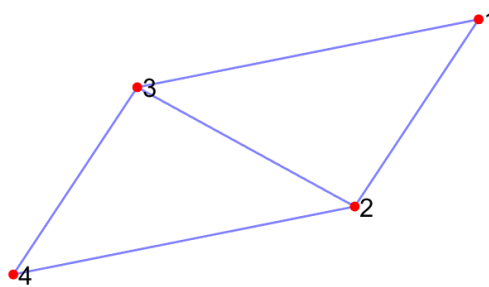
Obr. 2.25 – Multigraf

Prostý graf je graf neobsahující rovnoběžné hrany.



Obr. 2.26 – Prostý graf

Graf, který neobsahuje ani smyčky se nazývá *obyčejný graf*.



Obr. 2.27 – Obyčejný graf

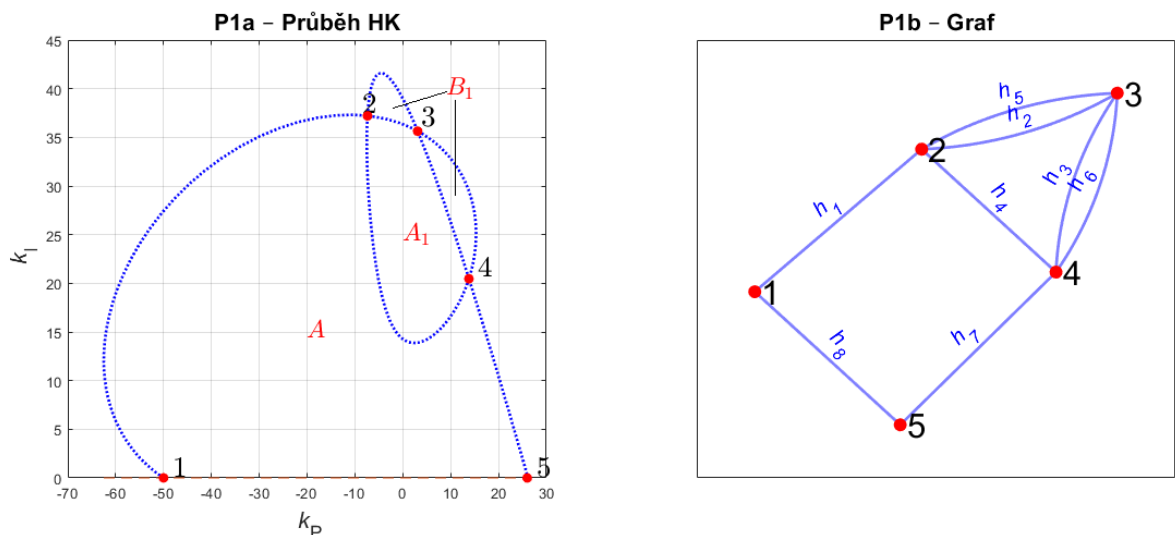
Reprezentace průběhu HK jako grafu

Průběh HK lze reprezentovat jako multigraf, kde průsečíky HK představují vrcholy resp. uzly multigrafu. Dále je, z hlediska následné rekonstrukce oblastí (viz pododíl 2.2.6), potřeba rozlišovat průsečíky HK se sebou samou – standardní uzly (*SU*) a průsečíky HK

s osou $k_I = 0$ – koncové uzly (KU). Z hlediska multigrafu jsou oba typy průsečíků, SU i KU, transparentní.

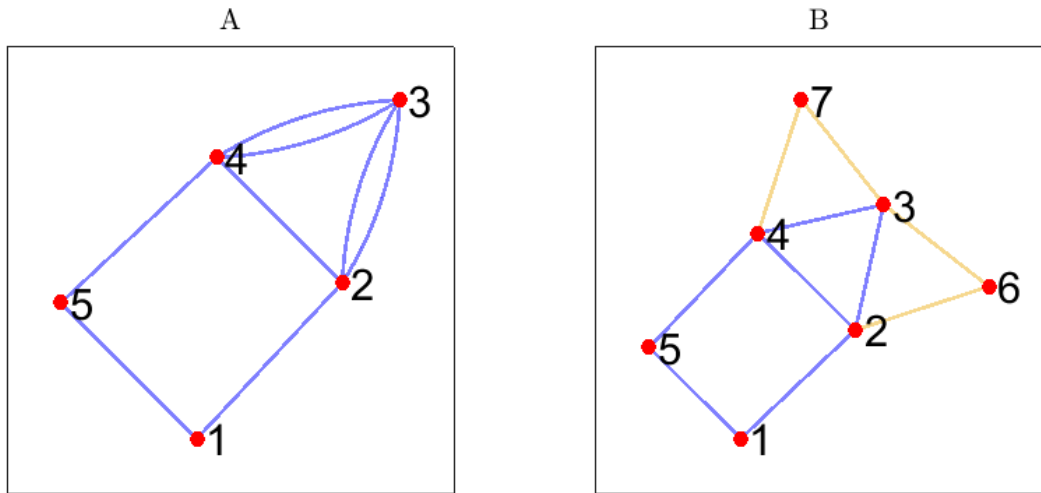
Jakmile je průběh hraniční křivky reprezentován prostřednictvím multigrafu, tak lze úlohu hledání jednotlivých oblastí převést na úlohu hledání tzv. *elementárních smyček* multigrafu. Úloha nalezení oblastí je úlohou nalezení elementárních smyček. *Elementární smyčka* multigrafu – označuje smyčku nebo podsmyčku, která již neobsahuje žádnou další podsmyčku. Lépe řečeno, elementární smyčka odpovídá posloupnosti vrcholů resp. hran, které definují oblast planárního grafu (viz níže v pododdílu 2.2.5).

Na základě znalosti elementárních smyček je možné oblasti oddělit a jednotlivě reprezentovat. Konstrukcí jednotlivých oblastí na základě znalosti elementárních smyček se zabývá pododdíl 2.2.6.



Obr. 2.28 – Hledané oblasti

Kromě multigrafu bylo v rámci SW realizace nutné pracovat i s reprezentací průběhu HK prostřednictvím obyčejného grafu. Důvod je ten, že mnoho funkcí MATLAB verze 2020a vyžaduje na svém vstupu obyčejný graf, který lze z jiných výše uvedených grafů získat funkcí `simplify`. Aby bylo možné hledat elementární smyčky v reprezentaci obyčejného grafu, tak bylo potřeba obyčejný graf *virtualizovat*. Virtualizace dvojitých hran multigrafu spočívala v nahrazení jedné z těchto hran novým virtuálním vrcholem a jeho spojením dvojicí virtuálních hran, viz obr. 2.29. Virtualizace oblasti typu smyčka spočívala v náhradě příslušné hrany dvojicí virtuálních vrcholů, které jsou spojeny trojicí virtuálních hran.



Obr. 2.29 – A – multigraf, B – odpovídající virtualizovaný obyčejný graf

Jakmile je obyčejný graf virtualizován, tak je na něm možné nalézt odpovídající elementární smyčky multigrafu. Elementární smyčka mezi vrcholy 3 a 4 multigrafu na obr. 2.29 – A je nalezena jako posloupnost vrcholů $V = (3, 4, 7)$ virtualizovaného obyčejného grafu na obr. 2.29 – B. Virtuální vrcholy jsou z pole, které reprezentuje elementární smyčku, následně odstraněny, mají totiž vyšší hodnotu, než je počet vrcholů multigrafu.

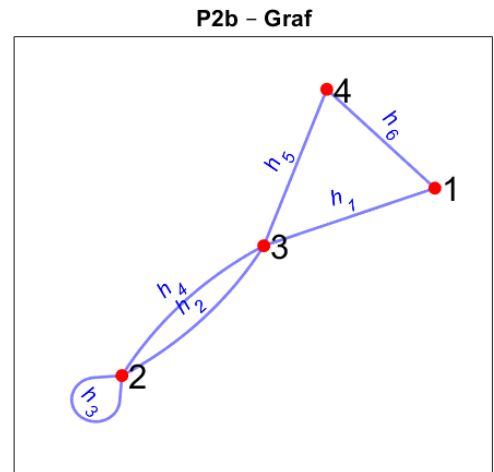
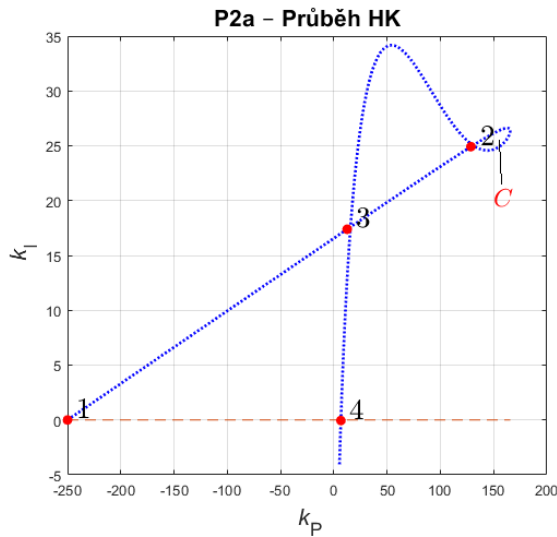
Z hlediska definované elementární smyčky lze definovat oblast typu:

Regulární mezi SU a KU – Jedná se o oblast, která je definována více než dvěma hranami a typicky vymezuje prostor mezi $k_1 = 0$ a HK (viz obr. 2.28 – A). Např. oblast A na obr. 2.28 je definována elementární smyčkou mezi vrcholy $V = (1, 2, 4, 5)$ resp. hranami (h_1, h_4, h_7, h_8) .

Regulární mezi SU – Opět se jedná o oblast, která má více než 2 hrany, v tomto případě však mezi standardními uzly (viz obr. 2.30 – A_1), typicky její reprezentace vede na kombinatorickou úlohu, elementární smyčka existuje mezi vrcholy $V = (2, 3, 4)$, v tomto případě ji však vymezují 4 možné sety hran $A_{1A} = (h_5, h_3, h_4)$, $A_{1B} = (h_5, h_6, h_4)$, $A_{1C} = (h_2, h_3, h_4)$, $A_{1D} = (h_2, h_6, h_4)$

Dvojitá hrana mezi SU – Jedná se o oblast, která je definována dvěma SU a stejným počtem hran (viz obr. 2.28 – B). Např. oblasti B na obr. 2.28 jsou definovány posloupnostmi vrcholů $V = (2, 3)$ a $V = (3, 4)$ resp. prostřednictvím posloupnosti hran (h_2, h_5) a (h_3, h_6) .

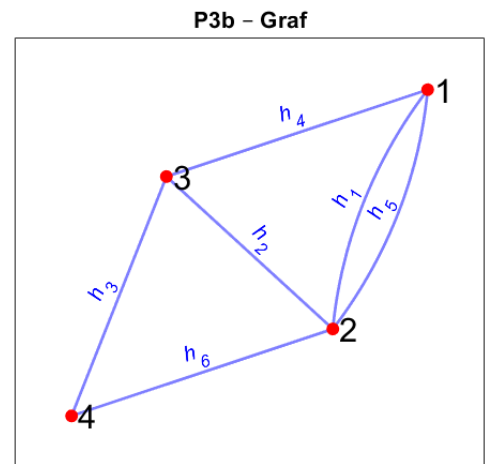
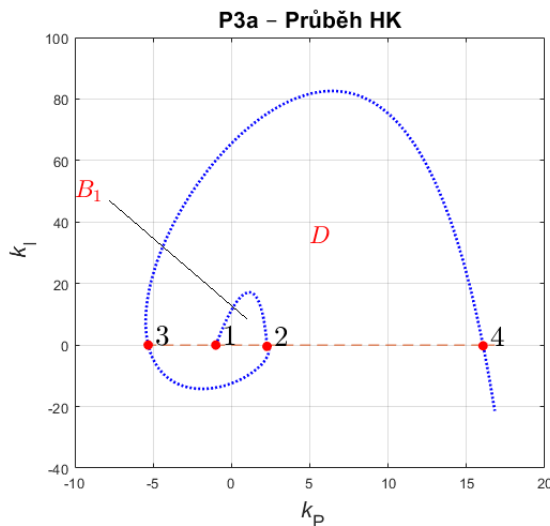
Smyčka – Jedná se o oblast, která je definována jednou hranou a jedním vrcholem, např. oblast C na obr 2.30, ta je definována vrcholem $V = 2$ a hranou h_3 .



Obr. 2.30 – Hledané oblasti

Dvojitá hrana mezi KU – Jedná se o oblast, která je definována dvěma KU a stejným počtem hran (viz obr. 2.31 – B_1), např. oblast B_1 na obr. 2.31 je definována vrcholy $V = (1, 2)$ a hranami h_1, h_5 .

Regulární mezi více než dvěma KU (viz obr. 2.31 – D) – Jedná se o nestandardní oblast, která byla objevena až po vytvoření obecnější verze programu, který konstruuje kompletní graf (nikoli jen graf mezi sousedními hodnotami ω_c).



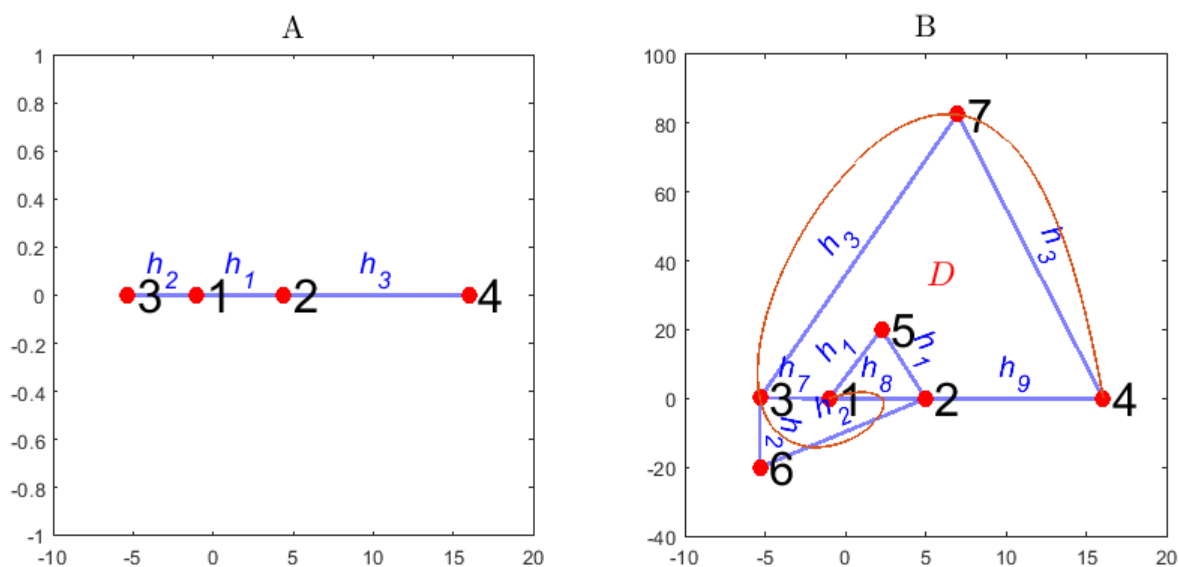
Obr. 2.31 – Regulární oblast mezi více než dvěma KU (D)

Významné je zjištění, že *reprezentace průběhu HK jako multigrafu není dostatečná*. Právě proto, že nezaručuje nalezení elementárních smyček pro všechny typy oblastí. Uvažujme oblast D na obr. 2.31 – P3a, ta je specifická tím, že ji nelze nalézt jako elementární smyčku obecného grafu a existuje mezi více KU. Tato oblast je definována posloupností vrcholů $V = (1, 2, 3, 4)$, resp. posloupností hran (proti směru hodinových ručiček (h_6, h_1, h_4, h_3)) (viz obr. 2.31 – P3b), ty však nesplňují definici elementární smyčky, protože touto smyčkou prochází hrana h_2 , která ji rozděluje na 2 elementární smyčky.

Podarilo se však nalézt způsob, pomocí kterého lze přístup reprezentace průběhu HK jako grafu používat nadále, pouze místo multigrafu je uvažován *planární graf (PG)*.

Planární graf

V případě planárního grafu jsou navíc *specifikovány souřadnice vrcholů* grafu v rovině (x, y) , resp. (k_P, k_I) . Např. graf na obr. 2.31 – P3a je sestaven tak, aby respektoval souřadnice průsečíků HK s osou $k_I = 0$.



Obr. 2.32 – A – planární graf mezi KU, B – odpovídající virtualizovaný planární graf mezi KU

Jelikož je v tomto případě planární graf na obr. 2.32 – A definován čtyřmi vrcholy, které leží na úsečce, tak je navíc nutné graf *virtualizovat* tzn. přidat virtuální vrcholy. Takto přidané virtuální vrcholy musí mít souřadnice, které odpovídají některému bodu průběhu HK mezi danými KU. Virtuální vrcholy jsou následně spojeny příslušnou zdvojenou virtuální hranou,

viz obr. 2.32 – B. Elementární smyčka oblasti D potom správně odpovídá posloupnosti hran (h_7, h_1, h_9, h_3) .

Hledání elementárních smyček jako oblastí planárního grafu

Právě objevením nového (Jiang et al., 1993) typu oblasti vznikla potřeba rozlišovat mezi obecným grafem a planárním grafem. *Planární graf je v práci využit pro hledání elementárních smyček*, na základě kterých jsou v pododdíle 2.2.6 jednotlivé oblasti reprezentovány.

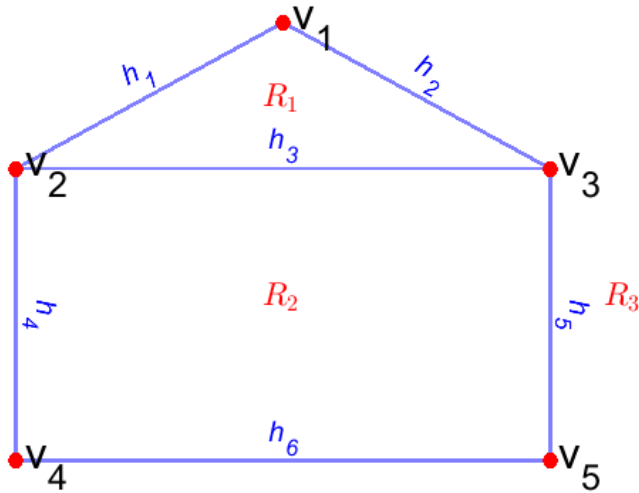
V obecném kontextu je kresba pomocí čar v podstatě planárním grafem a na vymezené plochy lze nahlížet jako na jeho oblasti. Z toho vyplývá, že chceme-li nalézt všechny vymezené plochy, tak musíme hledat oblasti planárního grafu (Jiang et al., 1993).

Graf má v daném prostoru tzv. ukotvení, v případě, že může být zakreslen tak, že se žádné hrany nekříží s výjimkou společných vrcholů. Graf je planární, pokud má ukotvení v rovině. Je dobře známo a dokázáno, že jakýkoli planární graf splňuje podmínku ukotvení v rovině, ve které jsou všechny hrany reprezentovány jako rovné čáry. V následující části je uvažována právě náhrada hran pomocí rovných čar (Jiang et al., 1993).

Planární graf dále rozděluje rovinu na oblasti (nebo také okna či stěny). Oblast planárního grafu je definována sadou hran, které tvoří jeho hranici, např. graf na obr. 2.33 má tři oblasti

$$\begin{aligned} R_1 &= ((v_1, v_2), (v_2, v_3), (v_3, v_3)), \\ R_2 &= ((v_2, v_4), (v_4, v_5), (v_5, v_3), (v_3, v_2)), \\ R_3 &= ((v_1, v_3), (v_3, v_5), (v_5, v_4), (v_4, v_2), (v_2, v_1)), \end{aligned} \tag{138}$$

kde R_3 se nazývá tzv. externí oblast, ta leží mimo graf a je nekonečná ve svém rozpětí (Jiang et al., 1993).



Obr. 2.33 – Ukázka planárního grafu

Mějme 2 hrany $e_1 = (v_i, v_j)$ a $e_2 = (v_j, v_k)$, předpokládejme, že nenarazíme na žádný další průsečík hrany s hranou v_j , když začneme otáčet hranu e_1 ve směru hodinových ručiček okolo v_j směrem k e_2 . Potom oblast vymezenou mezi e_1 a e_2 můžeme nazývat tzv. úhlovou oblastí. Dvě úhlové oblasti (v_i, v_j, v_k) a (v_j, v_k, v_l) lze nazvat styčnými. Je snadné určit, že graf o m hranách bude mít $2m$ úhlových oblastí. Uvažujme oblasti v grafu na obr. 2.33. Jelikož má planární graf 3 oblasti, tak úhlových oblastí bude právě 12, a to

$$\begin{aligned}
 &(v_3, v_1, v_2), (v_2, v_1, v_3), (v_1, v_2, v_3), (v_4, v_2, v_1), \\
 &(v_3, v_2, v_4), (v_2, v_3, v_1), (v_5, v_3, v_2), (v_1, v_3, v_5), \\
 &(v_2, v_4, v_5), (v_5, v_4, v_2), (v_4, v_5, v_3), (v_3, v_5, v_4).
 \end{aligned} \tag{139}$$

Oblast může být reprezentována sekvencí vrcholů, sekvencí hran a sekvencí úhlových oblastí. Např. oblast R_2 na obr. 2.33 může být reprezentována jako

$$\begin{aligned}
 &(v_2, v_4, v_5, v_3), \\
 &((v_2, v_4), (v_4, v_5), (v_5, v_3), (v_3, v_2)), \\
 &((v_2, v_4, v_5), (v_4, v_5, v_3), (v_5, v_3, v_2), (v_3, v_2, v_4)).
 \end{aligned} \tag{140}$$

Tyto reprezentace jsou ekvivalentní a lze je mezi sebou transformovat, tzn., že jakákoli z nich může být použita k reprezentaci oblasti (Jiang et al., 1993).

Pro nalezení oblastí bylo využito algoritmů publikovaných v (Jiang et al., 1993). Tyto algoritmy jsou popsány níže, a jejich implementace v programu MATLAB je převzata z (Matt, 2021).

Algoritmus 2. Zajišťující nalezení oblastí planárního grafu (resp. elementárních smyček) lze (dle (Jiang et al., 1993)) popsat následujícími obecnými kroky:

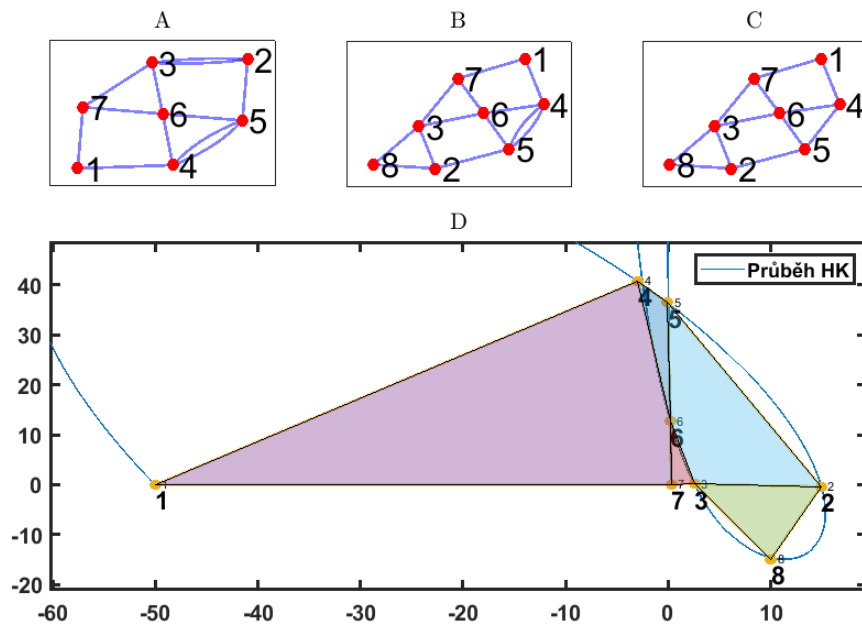
1. V rámci nalezení první oblasti zvol vrchol v_s jako počáteční vrchol. Výsledné oblasti budou reprezentovány posloupností vrcholů, které ji definují.
2. Mezi sousedními vrcholy s v_s nalezni vrcholy v_p a v_q , takové, že (v_p, v_s, v_q) tvoří úhlovou oblast. Označ v_p, v_s, v_q příslušně jako v_1, v_2, v_3 . Zaznamenej počáteční oblast (v_1, v_2, v_3) , a odstraň sousedící vazby $[v_1, v_2]$ a $[v_2, v_3]$ položením $A_{1,2} = 0$ a $A_{2,3} = 0$. Nastav $i = 0$.
3. Mezi sousedními vrcholy v_{i+2} nalezni v_{i+3} , takové, že $(v_{i+1}, v_{i+2}, v_{i+3})$ tvoří úhlovou oblast. Přidej v_{i+3} do listu reprezentující oblast a odstraň sousední vazby $[v_{i+2}, v_{i+3}]$ nastavením $A_{i+2, i+3} = 0$.
4. Pokud platí, že $v_{i+3} = v_1$, tak oblast byla nalezena a program pokračuje krokem 5. V opačném případě je i inkrementováno o 1 a program se vrací ke kroku 3.
5. V seznamu, reprezentujícím aktuální nalezenou oblast $(v_1, v_2, \dots, v_l, v_1)$, vyber první vrchol v_j , který splňuje $A_{j, j-1} = 1$, jako nový počáteční vrchol v_s a v_{j-1} jako v_q . Vrať se do kroku č. 2. Pokud žádný vrchol tuto podmínku nespĺňuje, tak pokračuj krokem č. 6.
6. Zkontroluj, jestli jsou úhly všech vrcholů nulové, pokud toto platí, tak ukonči program. V opačném případě vyber první vrchol s nenulovým úhlem jako nový počáteční vrchol v_s a v rámci nalezení další oblasti se vrať do kroku č. 2.

Výše uvedený postup, který byl publikován roku 1991, byl optimalizován, tento optimalizovaný postup však v rámci této práce není využit. Funkce zajišťující nalezení oblastí planárního grafu v následujícím algoritmu je `spatialgraph2D` a `polyshape` (Matt, 2021).

Algoritmus 3. Zajišťující sestrojení potřebných grafů a nalezení všech elementárních smyček, tedy funkce `getGraphs`, se dá popsat následujícími obecnými kroky:

1. Vstupem je normalizovaná matice *out* s indexy všech průsečíků HK a pole *xy* popisující průběh HK. Cílem funkce je nalézt všechny elementární smyčky (ES) a současně získat graf, který bude následně použit k sestrojení oblastí.
2. Na základě *out* získej souřadnice průsečíků *xy* a z nulových prvků na 3. řádce matice *out* urči matici *KU*. Označ a přeorganizuj *out1* = *out* pro potřeby sestrojení grafu.
3. Sestroj standardní multigraf (viz obr. 2.34 – A) a virtualizovaný graf (viz obr. 2.34 – B), který bude oproti standardnímu navíc obsahovat virtuální uzel a dvě virtuální hrany s ním spojené v místě, kde leží 2 KU vedle sebe. Pro oba grafy přidej hrany, které spojují KU, bude-li KU n potom hran mezi nimi bude $n - 1$.

4. V předchozích krocích byla také sestavována pomocná tabulka, která eviduje sekvenci přidávání hran do grafu a kompenzuje tak nedostatek programu MATLAB, který vrcholy řadí vzestupně bez ohledu na pořadí jejich přidání.
5. Na základě virtualizovaného grafu sestroj virtualizovaný obyčejný graf (viz obr. 2.34 – C) a na základě něj vytvoř virtuální planární graf (viz obr. 2.34 – D).
6. Pomocí funkcí nalezení oblastí planárního grafu `spatialgraph2D` a `polyshape` (Matt, 2021) najdi ES pro všechny typy regulárních oblastí na virtuálním planárním grafu.
7. Pomocí funkce `F_findEC_DH_L` detekuj vrcholy virtualizovaného grafu, které odpovídají ES oblasti typu dvojitá hrana či oblasti typu smyčka, ty přidej do matice *ES* k ostatním oblastem, je-li na nějakém řádku detekována duplicita, tak ji odstraň.
8. Výstupem funkce je multigraf *G*, matice indexů průsečíků křivky se sebou samou *out* a konečně matice *EC*, jejíž řádky jsou sestaveny z ES, tedy posloupnosti vrcholů, na základě grafu *G* (viz obr. 2.34 – A) a matice *EC* jsou následně reprezentovány jednotlivé oblasti.



Obr. 2.34 – A – multigraf, B – virtualizovaný multigraf (z hlediska hrany mezi KU – vrcholy 2 a 3), C – virtualizovaný obyčejný graf, D – virtualizovaný planární graf (z hlediska hrany mezi KU – vrcholy 2 a 3)

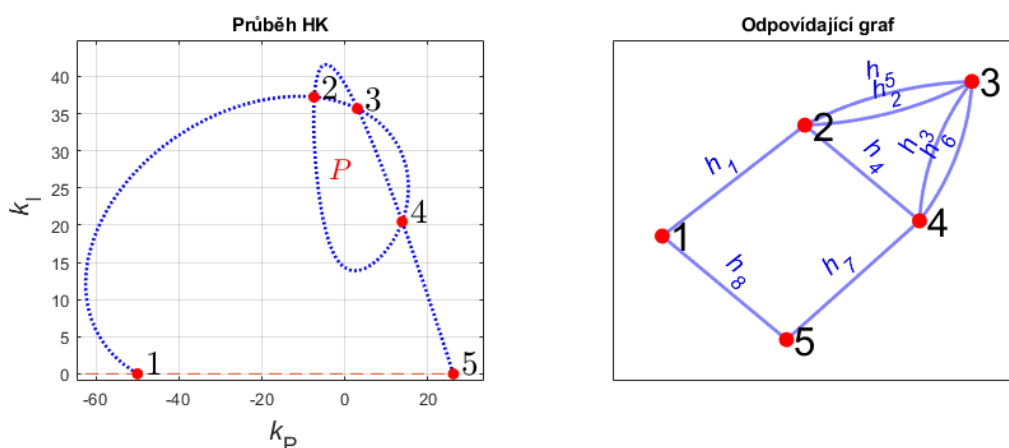
Matice nalezených elementárních smyček pro obr. 2.34 je

$$ES = \begin{pmatrix} 1 & 7 & 6 & 4 & 1 \\ 2 & 3 & 0 & 0 & 0 \\ 2 & 3 & 6 & 5 & 2 \\ 3 & 6 & 7 & 3 & 0 \\ 4 & 5 & 0 & 0 & 0 \\ 4 & 5 & 6 & 4 & 0 \end{pmatrix}. \quad (141)$$

2.2.6 Reprezentace jednotlivých oblastí

V předchozím pododdílu (viz 2.2.5) byl uveden princip získání elementárních smyček, které z hlediska posloupnosti vrcholů popisují jednotlivé oblasti. K reprezentaci průběhu HK je využit multigraf, který je, spolu s maticí elementárních smyček, výstupem předchozí části kódu. Pro nalezení elementárních smyček multigrafu bylo využito hledání oblastí (stěn) virtualizovaného planárního grafu.

Tento pododdíl se zabývá správnou konstrukcí jednotlivých oblastí, na základě již nalezených elementárních smyček. Je tedy potřeba vhodným způsobem navázat informaci o nalezené posloupnosti vrcholů na průběh hraniční křivky. A to tak, že na základě posloupnosti vrcholů v elementární smyčce je v multigrafu *nalezena odpovídající posloupnost hran*. Odpovídající SW realizace musí být zároveň dostatečně robustní, aby dokázala správně rozlišovat mezi různými typy oblastí a správně je oddělit.



Obr. 2.35 – Oblast P vedoucí na kombinatorickou úlohu

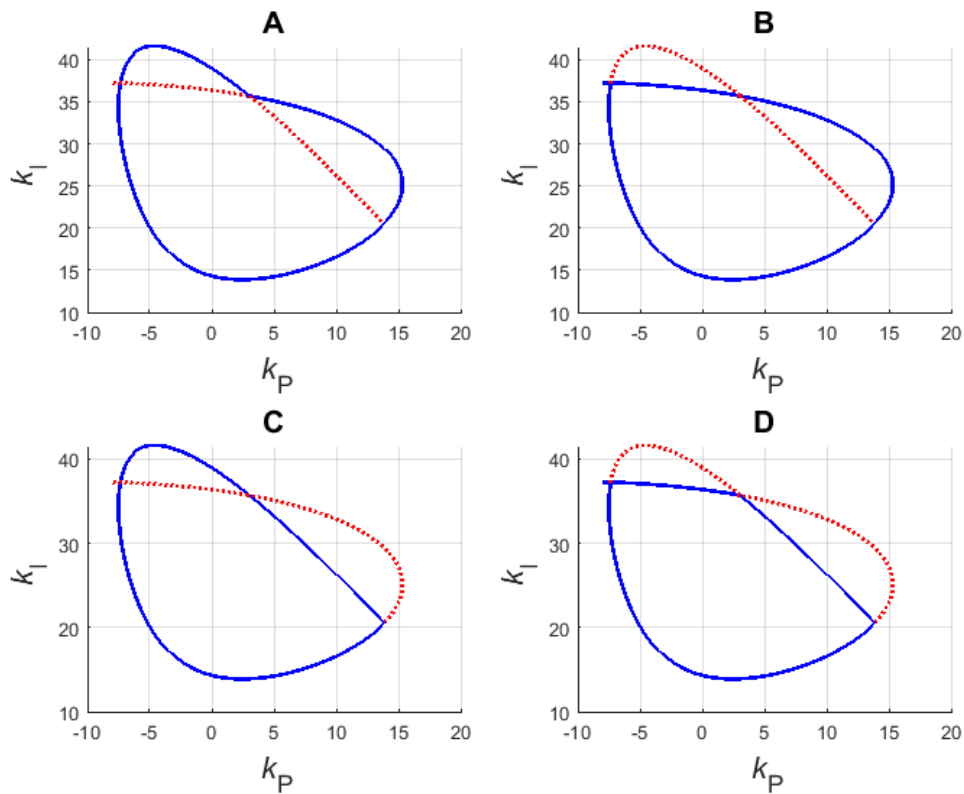
Důvod využití posloupnosti hran k oddělení oblastí je ten, že *samotné vrcholy neposkytují dostatečnou informaci o mezilehlé oblasti*. Tuto situaci ilustruje oblast P na obr. 2.35. Ta je reprezentována elementární smyčkou obsahující záznam o posloupnosti vrcholů $V = (2, 3, 4)$. Oblast mezi těmito vrcholy však není jednoznačná (viz obr. 2.36 – A ÷ D). Hledáním hran, které leží mezi příslušnými vrcholy $V = (2, 3, 4)$ v multigrafu je získána následující sada hran

$$\begin{aligned} H_A &= (h_5, h_3, h_4), \\ H_B &= (h_5, h_6, h_4), \\ H_C &= (h_2, h_3, h_4), \\ H_D &= (h_2, h_6, h_4). \end{aligned} \tag{142}$$

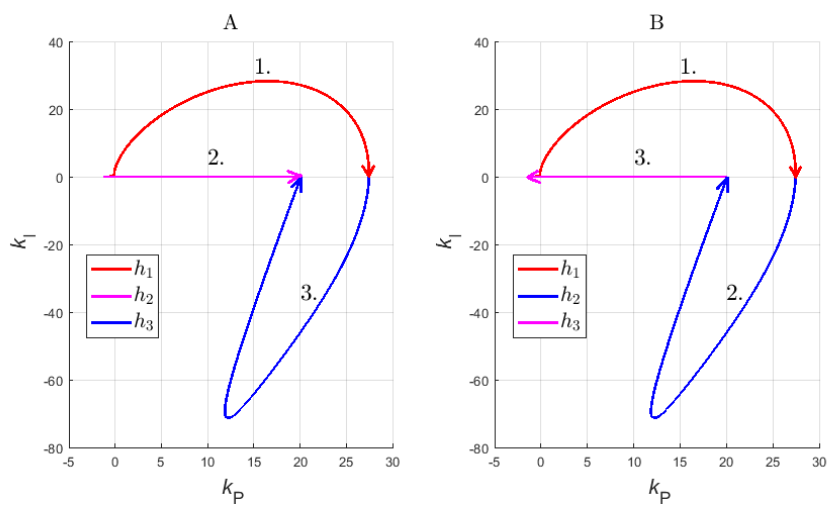
Počet těchto kombinací je dán 2^n , kde n je počet dvojitých hran mezi danými vrcholy. Oblasti odpovídající různým kombinacím hran (142) jsou znázorněny na obr. 2.36 – A ÷ D.

V souvislosti s možnými sety kombinací hran je potřeba řešit úlohu *výběru správné oblasti z množiny možných oblastí*. Toho lze dosáhnout výpočtem obsahu pro každou z těchto oblastí, přičemž správnou reprezentací oblasti je ta, která má nejmenší obsah (ostatní oblasti jsou již odděleny předchozí částí kódu, jsou tedy vyřazeny takové kombinace hran aktuální oblasti, které by s nimi měli společný průnik). Tímto způsobem je získána oblast na obr. 2.36 – D.

V souvislosti s výpočtem obsahu je však potřeba řešit další dílčí podproblém. Aby byl obsah oblasti stanoven správně, tak je potřeba *zajistit správnou orientaci jednotlivých segmentů HK* , které jsou reprezentovány hranami multigrafu (resp. jejich návaznost a směr). Také záleží v jakém pořadí jsou tyto segmenty přidávány do výsledného pole, které reprezentuje hranice oblasti. Oblast je správně rekonstruována pokud její hranice tvoří uzavřený cyklus (což $l(\omega)$ pro narůstající ω nemusí zaručit), viz obr. 2.37 – B. Obsah je stanoven jako plocha pod křivkou, je tedy ještě potřeba zajistit, že se od plochy nad osou $k_I = 0$ neodečte ta, která leží pod $k_I = 0$. Toho lze snadno docílit např. $k_I = k_I + c$, kde $c < \infty$.



Obr. 2.36 – Oblasti odpovídající různým kombinacím hran (142)



Obr. 2.37 – A – špatná orientace a pořadí segmentů křivky, B – správná orientace a pořadí segmentů křivky

Algoritmus 4. Funkce `getRegions` zajišťující správné oddělení jednotlivých oblastí funguje následovně:

1. V matici xy_n vytvoř nové segmenty křivky pro dvojice uzlů, které odpovídají posloupnosti KU. Označ ec jako příslušný řádek matice EC .
2. Pro všechna $i = 1, \dots, n - 1$, kde n je délka pole ec :
 1. urči hranu mezi $ec(i)$ a $ec(i + 1)$ a tu zaznamenej jako $edg(i)$.
3. Vytvoř E , jako pole které bude obsahovat sekvenci hran, které leží mezi stejnými uzly a e jako pole obsahující hrany, které nejsou duplicitní z pohledu dvojice uzlů grafu.
4. Je-li E neprázdné, tak na základě E stanov všechny možné kombinace hran E_{CA} , na konec každého řádku této matice přidej e .
5. Pro všechna $j = 1, \dots, m$, kde m je počet řádků matice E_{CA} volej funkci `edgesSequentially` pro daný řádek matice E_{CA} . Funkce `edgesSequentially` je popsána níže.
6. Vyber tu permutaci, pro kterou je obsah odpovídající oblasti nejmenší a pole xy_f označ jako výstup funkce, stejně tak označ n_{idx} jako sekvenci indexů vrcholu dané oblasti, viz např. obr. 2.36 – D.

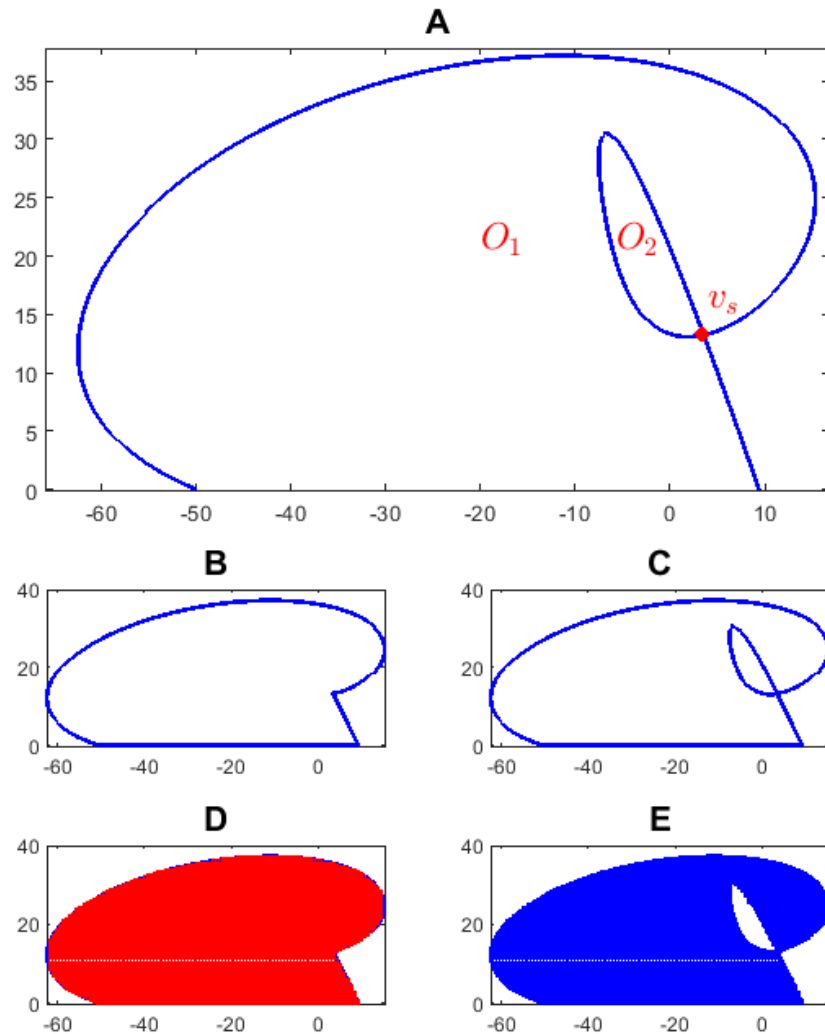
Algoritmus 5. Funkce `edgesSequentially`, zajišťující správné pořadí a orientaci segmentů HK, které reprezentují hranici oddělené oblasti, funguje následovně:

1. Funkci je poskytnut řádek matice E_{CA} , pole xy popisující průběh HK, xy_n s nově vytvořenými segmenty křivky pro spojení dvojice KU a matice tab , která odpovídá grafu a nese informaci o rozmezích indexů, které odpovídají segmentům HK, resp horizontální osy. Pro všechna $k = 1, \dots, m$:
 - a) Na základě k -tého záznamu z tab proved konstrukci k -tého segmentu hraniční křivky na základě xy a tu ulož do xy_t .
 - b) V případě, že se jedná o hraniční křivku, která se nachází mezi koncovými uzly, tak proved konstrukci na základě xy_n a výsledek ulož do xy_t .
2. Pro všechna $k = 1, \dots, m$:
 - a) Na základě xy_t urči koncovou x, y souřadnici koncového bodu xy_{ref} daného segmentu křivky a pro všechna $l = k + 1, \dots, m$:
 - i. Urči xy_1 jako počáteční bod l -tého segmentu křivky a xy_2 jako koncový bod l -tého segmentu křivky (resp jejich souřadnice).
 - ii. Vypočítej euklidovskou vzdálenost $e1$ mezi xy_{ref} a xy_1 a euklidovskou vzdálenost $e2$ mezi xy_{ref} a xy_2 .
 - b) Vyber tu hranu a její orientaci, pro kterou je $e1$ resp. $e2$ nejmenší a v případě, že je $e1 < e2$, tak daný segment křivky převrať pomocí MATLAB funkce `flip`. Zařaď nalezenou hranu za aktuálně vyšetřovanou.
3. Na základě pole xy_f , které odpovídá správné posloupnosti a orientaci hran stanov také pole n_{idx} obsahující indexy vrcholů dané oblasti.
4. Výstupem je oblast se správným pořadím a orientací segmentů křivky, které svým průběhem tvoří uzavřený cyklus, viz např. situace na obr. 2.37 – B. Výstupem je také pole n_{idx} , které je využito v následujících částech kódu.

2.2.7 Zajištění topologického uspořádání oblastí

V rámci předchozích kroků bylo zajištěno oddělení jednotlivých oblastí, ukazuje se však, že pro některé raritní případy může nastat situace, že *oddělená oblast zahrnuje nějakou další oddělenou oblast*.

Uvažujme průběh HK na obr. 2.38 – A, je zřejmé, že tento průběh rozdělil rovinu (k_P, k_1) na oblast O_1 a O_2 . Oblast O_2 , je po předchozích krocích algoritmu reprezentována korektně. Problém však nastává s reprezentací oblasti O_1 . Dosavadní algoritmus sice zajistí oddělení jednotlivých oblastí, nemá však informaci o uspořádání oblastí z hlediska jejich vzájemné inkluze. Důsledkem je chybná reprezentace hranice oblasti O_1 , kterou zachycuje obr. 2.38 – B.

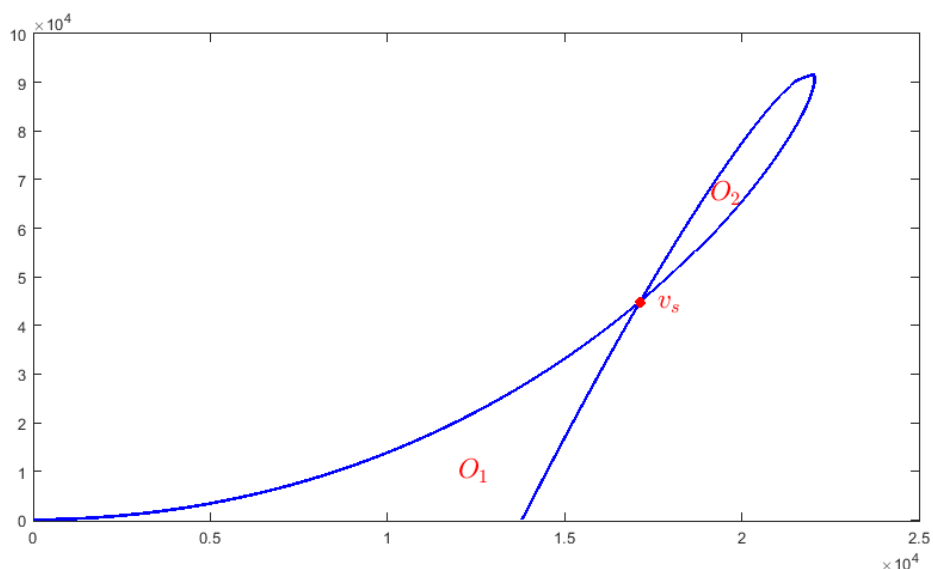


Obr. 2.38 – A – průběh HK vedoucí na chybnou reprezentaci vymezených podoblastí, B – chybná reprezentace hranice oblasti O_1 a jí odpovídající chybné vyplnění (D), C – správná reprezentace hranice oblasti O_1 a jí odpovídající správné vyplnění (E)

S nesprávnou reprezentací resp. zobrazením oblastí se váže praktický problém, který je v zásadním rozporu s principem metody – výsledkem je oblast, která může zahrnovat jak stabilní tak nestabilní část (podoblast). Označení všech vnitřních bodů chybně reprezentované oblasti O_1 zachycuje obr. 2.38 – D. V rámci zajištění správné reprezentace oblasti O_1 je potřeba vyjmout oblast O_2 , která se celá nachází uvnitř O_1 , viz obr. 2.38 – E. Správné určení vnitřních bodů (pomocí přístupu vektorové grafiky, kterým se zabývá pododdíl 2.2.8) oblasti O_1 lze zajistit reprezentací hranice oblasti O_1 tak, že bude zahrnovat i hranici oblasti O_2 , viz obr. 2.38 – C. Tímto je možné následně získat správnou reprezentaci oblasti O_1 , kterou zachycuje obr. 2.38 – E.

Na základě výše uvedeného lze vyslovit následující předpoklad. Tato situace nastane, v případě, že oblast leží uvnitř jiné takovým způsobem, že s vnější oblastí má *jeden společný vrchol*. Dalo by se říci, že se jedná o nový typ oblasti:

Oblast nacházející se celá uvnitř jiné oblasti, se kterou má společný vrchol – Jedná se o typ oblasti definované ve vztahu k jiné již oddělené oblasti. Podobným způsobem by se chovala oblast „typu smyčka“, ta má s jinou oblastí taktéž společný vrchol, nicméně oblast typu smyčka se typicky nachází mimo oblast, se kterou sdílí vrchol, viz obr. 2.39. To však nemusí platit vždy (viz obr. 2.38 – A). Takto může mít společný vrchol s vnější oblastí také některá již dříve definovaná oblast, např. oblast typu „dvojitá hrana“ (viz obr. 2.40 – B) či oblast „regulárního typu“ (viz obr. 2.46 – R_5).



Obr. 2.39 – Průběh HK definující vnější oblast typu smyčka

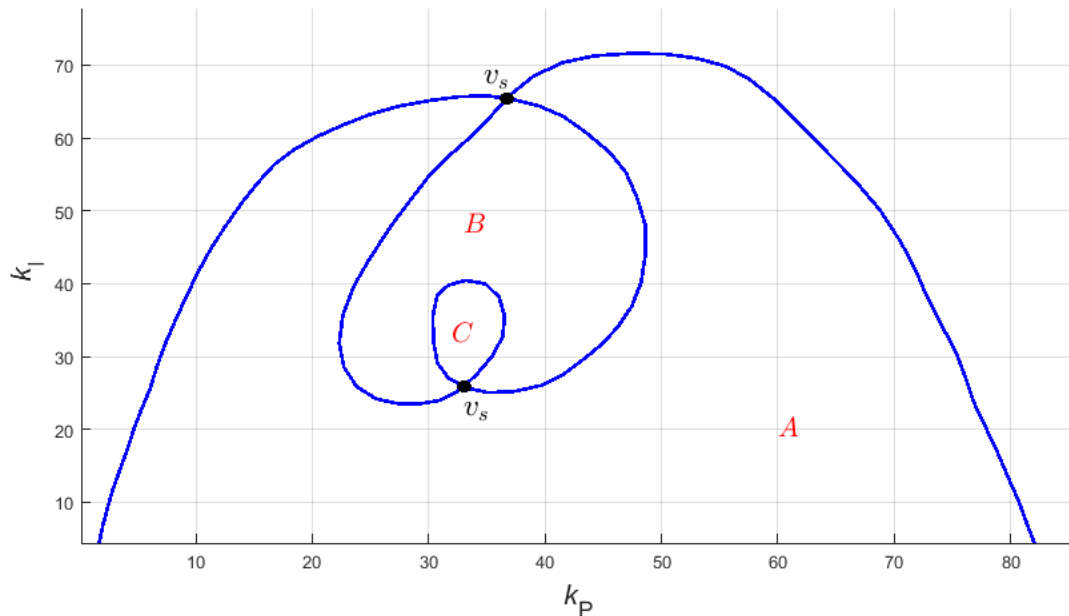
Poznámkou je vhodné uvést následující. V případě, že má oblast s jinou oblastí společné alespoň 2 vrcholy, tak je oddělení provedeno již na úrovni předchozích částí algoritmu. Např. je taková mezilehlá oblast označena jako „oblast typu dvojitá hrana“ a případné oddělení od ostatních je provedeno na úrovni minimalizace obsahu vymezené plochy, kterým se zabývá pododíl 2.2.6.

Předchozí část algoritmu, využívající poznatků teorie grafů k reprezentaci oblastí, umožňuje od sebe jednotlivé oblasti oddělit. Oddělené oblasti však stále mohou být *součástí jiných oddělených oblastí*, viz např. obr. oblast O_2 , která je součástí oblasti O_1 na obr.

2.38 – D. Mezi některými dvojicemi oblastí lze definovat relaci \prec , která má ten význam, že pokud $A \prec B$, pak A leží celá uvnitř B . Obdobně lze definovat relaci \succ , která má ten význam, že pokud $A \succ B$, pak A zahrnuje i celou B . Seřazení oblastí tak, aby vždy platilo $A_i \prec A_j$ tehdy, jestliže $i < j$ a relace \prec je mezi A_i a A_j definovaná, se nazývá *topologické uspořádání*.

Uvažujme například jednotlivé oblasti na obr. 2.40, předchozím postupem jsou jednotlivé oblasti odděleny jako obr. 2.41 – A, B, C, je zřejmé, že oblast A zahrnuje současně oblast B i C, a oblast B zahrnuje i oblast C, to se dá zapsat jako

$$C \prec A, B \prec A, C \prec B. \quad (143)$$



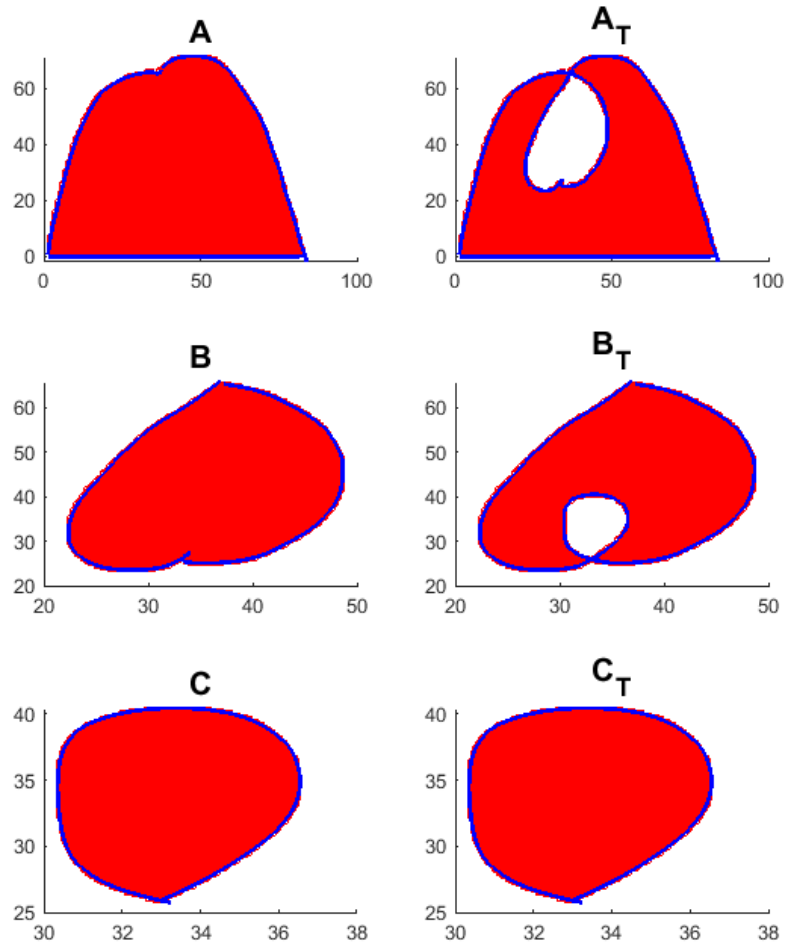
Obr. 2.40 – Průběh HK zahrnující nový typ oblasti z hlediska topologie

Oblasti, pro které je definovaná relace \prec , je možné seřadit tak, aby platilo $A_1 \prec A_2 \prec \dots \prec A_n$ pokud $A_1 \prec A_2, A_2 \prec A_3, \dots, A_{n-1} \prec A_n$. Při zobrazení oblasti A_k je nutné vyjmout část plochy, která odpovídá A_{k+1} v seřazení (pokud taková existuje), přičemž další oblasti v pořadí se neuvažují. Pro oblasti na obr. 2.40 platí

$$A \succ B, B \succ C, \quad (144)$$

tedy oblast B v uspořádání předchází oblast A, to, že jiná oblast předchází oblast B, není podstatné. Zobrazením oblasti B (zahrnutím segmentů HK) v rámci A vznikne oblast

2.41 – A_T . Podobně zobrazením oblasti C v rámci oblasti B vznikne 2.41 – B_T . Oblast 2.41 – C_T již v uspořádání nemá předcházející oblast, je tedy zobrazena korektně.

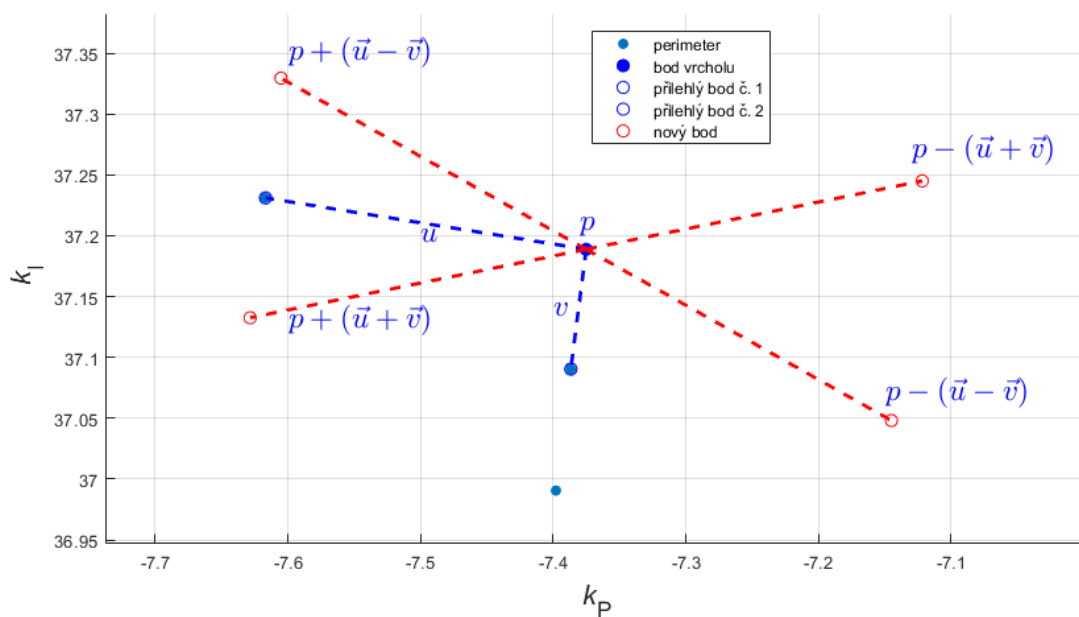


Obr. 2.41 – Nesprávná (A, B, C) a správná (A_T, B_T, C_T) reprezentace jednotlivých oblastí, které vymezi průběh HK na obr. 2.40

V rámci zjištění vzájemné relace mezi oblastmi jsou uvnitř každé oblasti poblíž její hranice generovány testovací body, u nichž se následně testuje příslušnost do ostatních oblastí. Je-li v rámci oblasti zjištěna i příslušnost této oblasti do oblasti jiné, pak je příslušným způsobem tato relace zaznamenána a nakonec je provedena jejich správná reprezentace z hlediska dat, které je popisují.

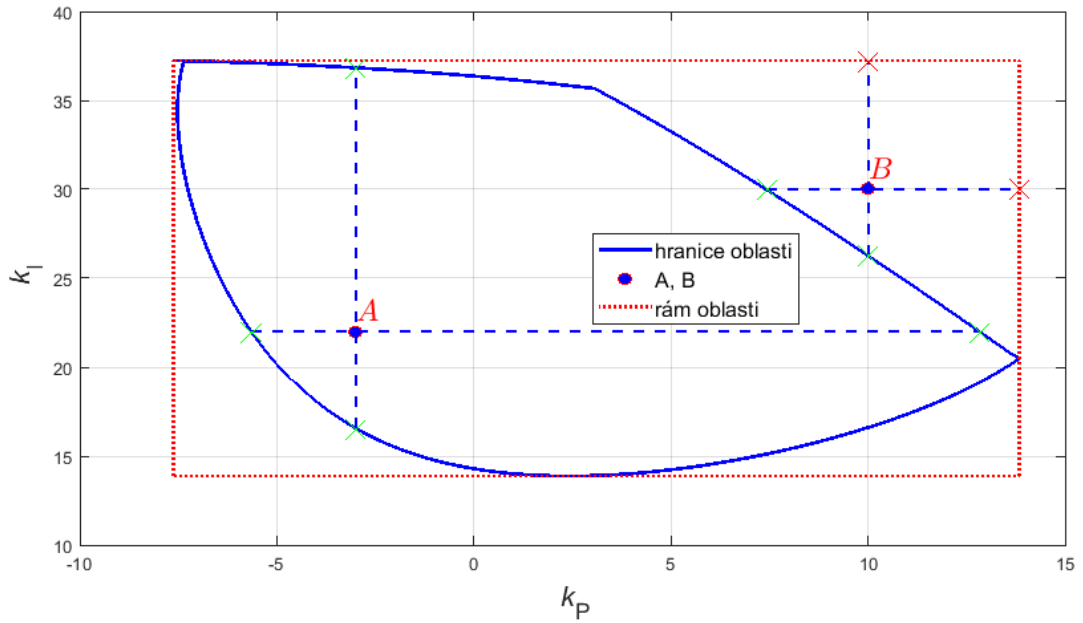
Obecně lze postup zajišťující správnou reprezentaci oblastí, tedy takovou, která respektuje topologické uspořádání oblastí, popsat následovně:

1. Nejprve se v poli, které reprezentuje hranici oblasti, určí n referenčních bodů, přičemž body, které popisují vrcholy, jsou vždy součástí těchto referenčních bodů. Dále se určí *nové body* (viz obr. 2.44 – A), které se vypočítají jako kladné a záporné vektorové součty předcházejícího a následujícího bodu vzhledem k referenčnímu. Princip výpočtu nových bodů ilustruje obr. 2.42.

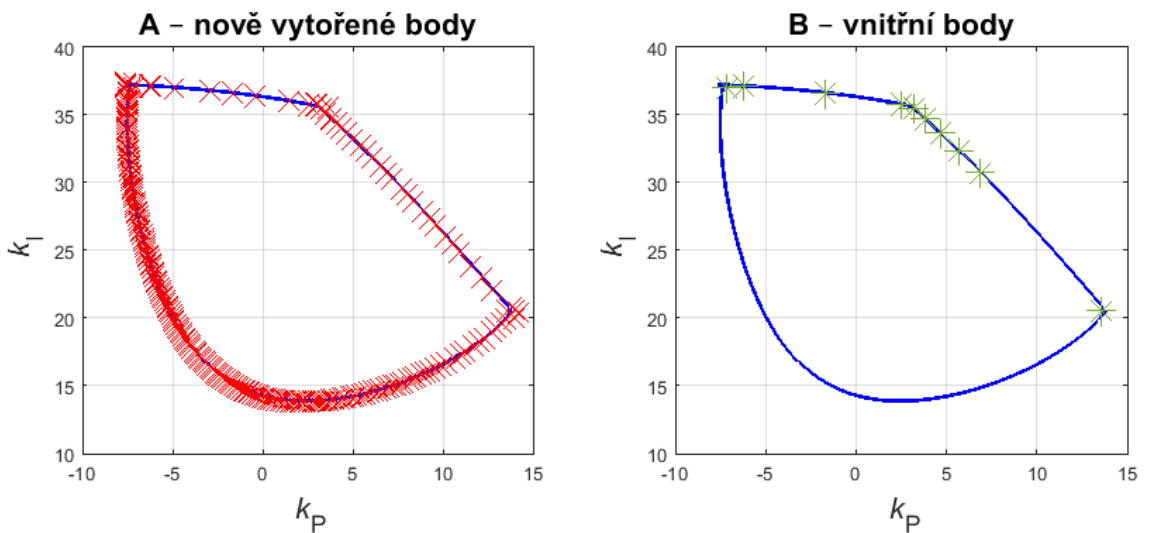


Obr. 2.42 – Určení nových bodů na základě vektorového součtu

2. Dále je z množiny nových bodů vybráno m vnitřních bodů (viz obr. 2.44 – B), tedy bodů, které leží uvnitř oblasti. Princip nalezení *vnitřních bodů*, který zachycuje obr. 2.43, je následující. Z referenčního bodu jsou vedeny 4 přímky, následně jsou hledány průsečíky těchto přímek s hranicí oblasti. Bod je označen jako vnitřní v případě, že ani jedna ze souřadnic průsečíků nenáleží rámu oblasti. Takto má každá oblast m vnitřních bodů, které ji definují.

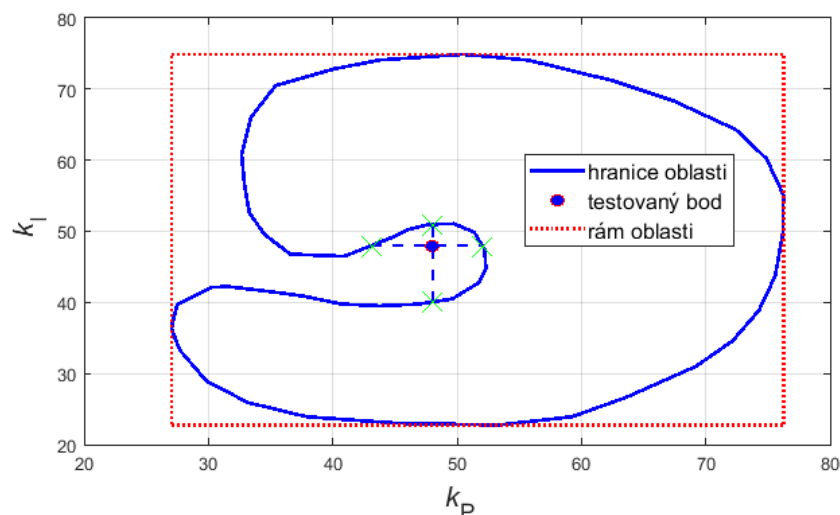


Obr. 2.43 – Princip určení vnitřních bodů z množiny nových bodů na vstupu funkce



Obr. 2.44 – Výběr m vnitřních bodů z množiny vypočítaných nových bodů

Tato jednoduchá a velice rychlá metoda pro určení vnitřních bodů má své omezení – uvažujme hypotetickou oblast na obr. 2.45 a testovací bod mimo oblast, je zřejmé, že všechny průsečíky jednotlivých přímek s hranicí oblasti budou existovat a tento bod bude chybně označen jako vnitřní bod. Nicméně tato situace je velice nepravděpodobná.

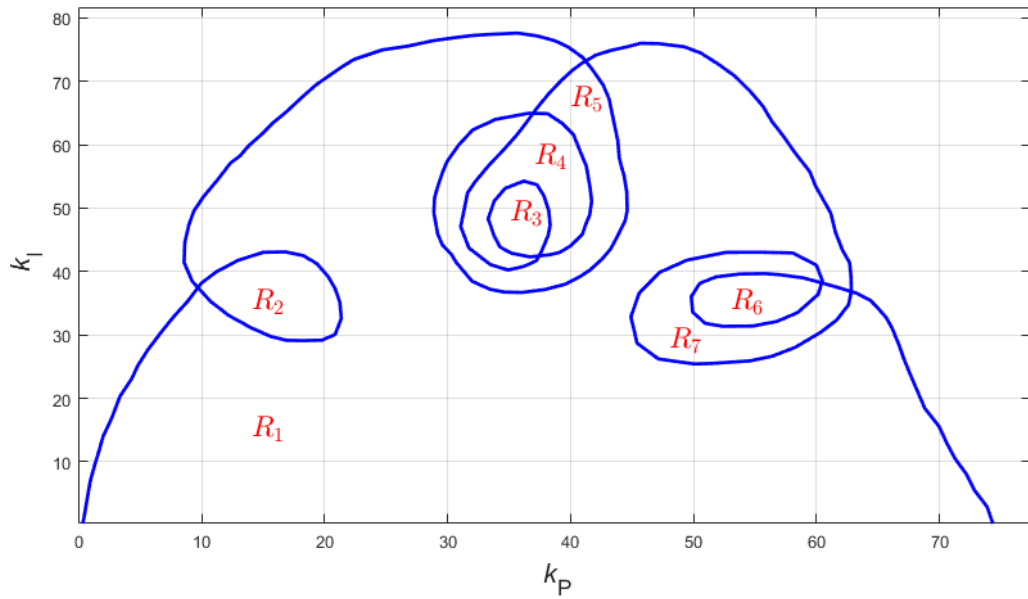


Obr. 2.45 – Testovaný bod chybně označený jako vnitřní bod

Alternativně lze pro určení vnitřních bodů funkci rozšířit ve smyslu výpočtu úhlu mezi vyšetřovaným bodem a všech po cestě ležících bodech popisujících hranice oblasti, k tomu lze využít modifikovanou funkci atan vracející úhel v intervalu $(0, 2\pi)$, leží-li bod uvnitř oblasti, měl by úhel odpovídat 2π . S tím se však váže zvýšená výpočetní náročnost, která roste s druhou mocninou vyšetřovaných dvojic.

3. Jakmile je každá oblast definována množinou vnitřních bodů, je možné se ptát, zda-li vnitřní body oblasti A nejsou zároveň vnitřními body oblasti B . Ke zjištění, zda-li jsou vnitřní body zároveň vnitřními body oblasti jiné, lze využít stejné metody, která byla využita k nalezení vnitřních bodů obecně. Pouze místo hranice oblasti, které dané vnitřní body náleží, se uvažují hranice všech oblastí ostatních. Tímto způsobem je vytvořena tabulka, resp. matice (viz např. (146)), která zachycuje *topologické uspořádání oblastí*, na základě kterého je následně možné zajistit *správnou reprezentaci* jednotlivých oblastí.

Příklad – uvažujme hypotetický systém č. 1 s průběhem HK na obr. 2.46.



Obr. 2.46 – Průběh HK hypotetického systému a znázornění jednotlivých vymezených oblastí

Na základě výše uvedených kroků je konstruována matice

$$T_A = \begin{pmatrix} & R_1 & R_2 & R_3 & R_4 & R_5 & R_6 & R_7 \\ R_1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ R_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_4 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ R_5 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ R_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_7 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad (145)$$

která obsahuje záznam o relaci mezi oblastmi z hlediska jejich vzájemné inkluze. Přičemž jednička je na pozici (i, j) pokud $R_j \prec R_i$, tedy pokud všechny vnitřní body oblasti R_j

jsou zároveň uvnitř hranice oblasti R_i . Následně jsou odstraněny záznamy již obsažené v rámci oblasti, která v zobrazení předchází

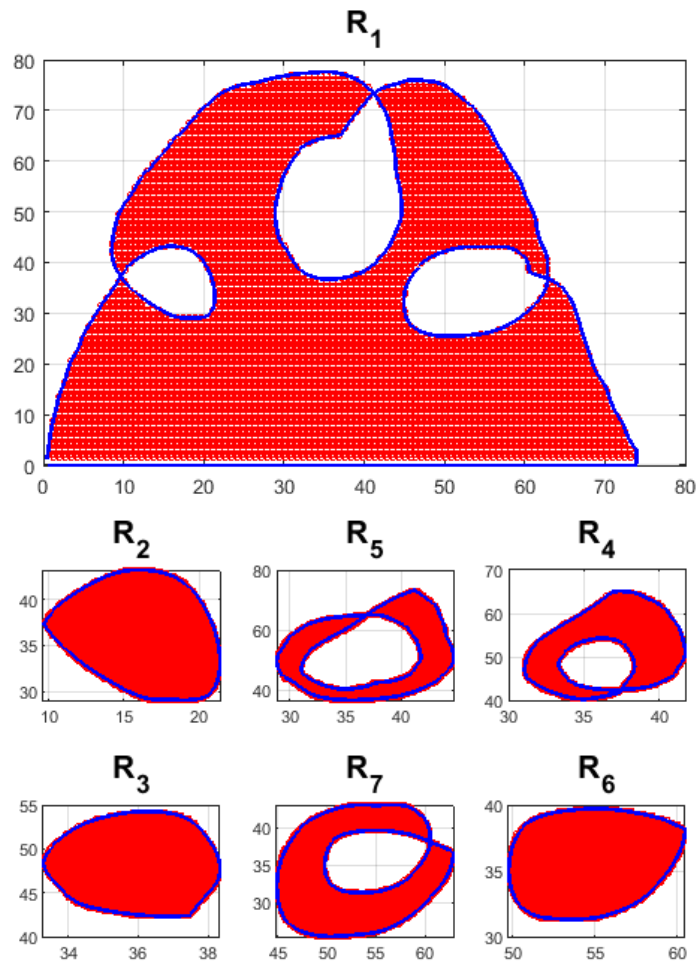
$$T_A = \begin{pmatrix} & R_1 & R_2 & R_3 & R_4 & R_5 & R_6 & R_7 \\ R_1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ R_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_4 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ R_5 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ R_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_7 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad (146)$$

Toho je docíleno tak, že pro řádek i jsou prohledány všechny sloupce všech ostatních řádků. V případě, že sloupec k řádku j obsahuje záznam ve stejném sloupci jako vyšetřovaný řádek i , tak $T_A(i, k) = 0$. Způsob, jakým došlo k sestavení matice (146) detailně zachycují následující kroky

$$\begin{aligned} T_{A1} &= \begin{pmatrix} 0 & 1 & \mathbf{1} & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, T_{A2} = \begin{pmatrix} 0 & 1 & \mathbf{0} & \mathbf{1} & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \\ T_{A3} &= \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \end{pmatrix}, T_{A4} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \end{aligned} \quad (147)$$

kde T_{A1} představuje následující situaci. Uvažujme, že v rámci zobrazení oblasti R_1 je potřeba zobrazit zároveň R_4 , platí tedy $R_4 \prec R_1$, není však potřeba, jak již bylo zmíněno dříve, zobrazovat i oblasti, které zahrnuje oblast předcházející v uspořádání, v tomto případě

oblast R_3 – odstraňme tedy z prvního řádku všechny záznamy (červeně), které obsahuje řádek čtvrtý (modře). Dále pro T_{A2} platí, že $R_5 \prec R_1$, v rámci zobrazení oblasti R_1 však stačí zobrazit R_5 , podoblasti R_5 (jmenovitě R_3 a R_4) jsou tedy ze záznamu prvního řádku odstraněny. Dále lze postupovat obdobně, za zmínku stojí také krok T_{A4} , pro zobrazení R_5 platí, že $R_3 \prec R_4 \prec R_5$, ze záznamu je tedy odstraněna oblast (červeně), která je již součástí R_4 (modře), z čehož vyplývá, že výsledek nezávisí na pořadí vyšetřovaných oblastí. Výsledné oblasti respektující topologii jsou na obr. 2.47.

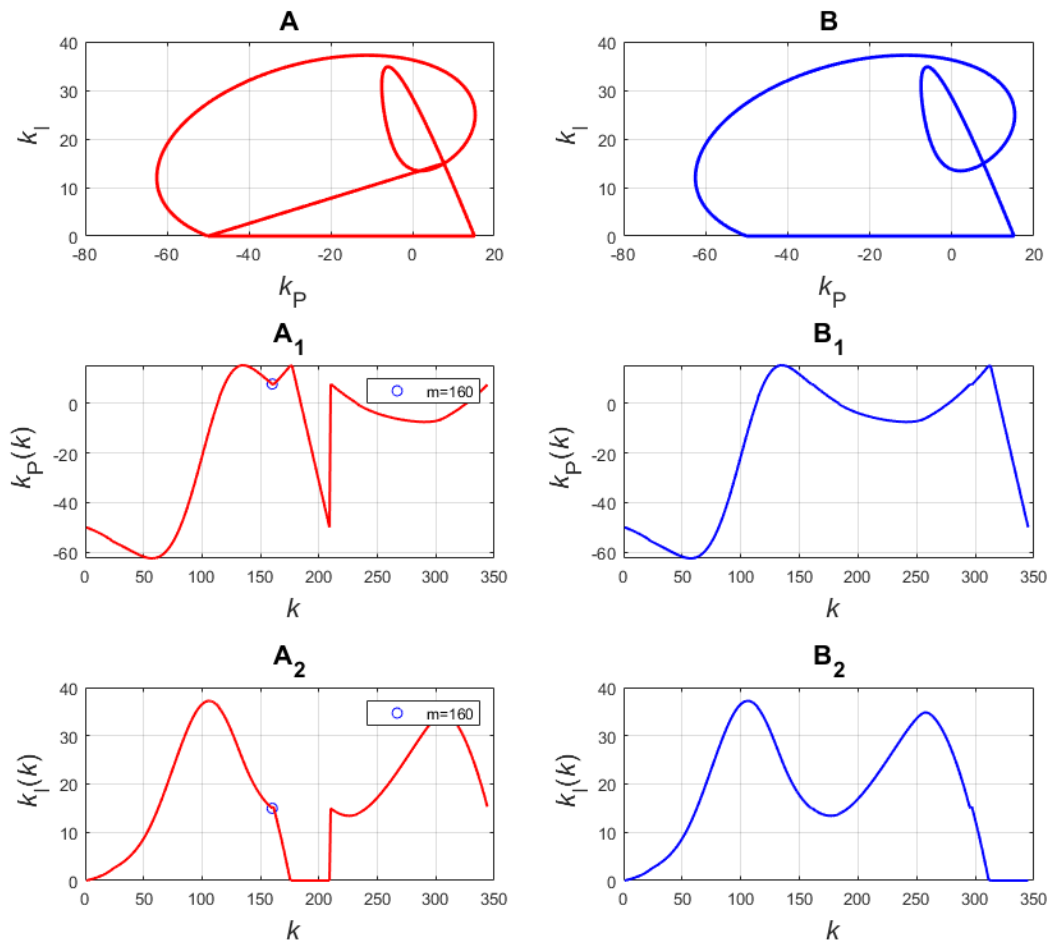


Obr. 2.47 – Správné zobrazení jednotlivých oblastí hypotetického systému č. 2

Samostatnou částí algoritmu je také funkce, která má zajistit správnou reprezentaci oblasti, a spočívá ve *správném spojení vektorů* popisujících hranice spojovaných oblastí. Nově přidávanou oblast je potřeba spojit se stávající v místě jejich společného vrcholu, který představuje konkrétní hodnota indexu v poli popisujícím průběh hranice oblas-

tí. V opačném případě dojde ke skokové změně, v jejímž důsledku nebude, postupem uvažovaným v pododdíle 2.2.8, zajištěno správné vyplnění oblasti body.

Nesprávné (A) a správné (B) spojení vektorů, jejichž výsledkem je nová oblast, zachycuje obr. 2.48. Přičemž A_1, A_2 zachycuje skok na indexu $k = 160$. B – zachycuje správnou reprezentaci oblasti, kde B_1, B_2 zachycuje průběh, ve kterém ke skokové změně nedochází – pole popisující menší oblast je správně vloženo do pole popisující oblast větší.



Obr. 2.48 – A a související A_1, A_2 zachycují špatnou reprezentaci z hlediska dat, B a související B_1, B_2 zachycují správnou reprezentaci z hlediska dat.

Jak již bylo zmíněno výše, nejprve je potřeba získat nové vnitřní body oblasti

Algoritmus 6. Zajišťující získání nových bodů podél hran definujících oblast se dá v obecné rovině popsat jako:

1. Rozšiř pole n_{idx} popisující indexy dané oblasti náležející vrcholům o dalších n hodnot, přičemž zajisti, aby původní n_{idx} byl na začátku. Následně pro všechny $i = 1, \dots, n$, kde n je délka n_{idx} :
 - a) Z výchozího pole xy popisující průběh HK, urči „cyklické“ pole směrem doprava $xy_p = xy(n_{idx}), \dots, xy(n), 1, \dots, xy(n_{idx} - 1)$ a „cyklické“ pole směrem doleva $xy_l = xy(n_{idx}), \dots, 1, 1, xy(n), \dots, xy(n_{idx} + 1)$. Urči souřadnici $xy_{act} = xy(i)$.
 - b) Pro všechna $j = 1, \dots, n$:
 - i. Urči $xy_{nxt} = xy_r(j)$ a v případě, že $xy_{nxt} \neq xy_{act}$, tak cyklus ukonči.
 - c) Pro všechna $j = 2, \dots, n$:
 - i. Urči $xy_{prv} = xy_l(j)$ a v případě, že $xy_{prv} \neq xy_{act}$, tak cyklus ukonči.
 - d) Na základě určeného xy_{act} (představuje bod p), xy_{nxt} (představuje KB vektoru \vec{v}) a xy_{prv} (představuje KB vektoru \vec{u}) stanov 4 varianty vektorového součtu viz obr. 2.42, ty ulož do $NB(i)$.

Algoritmus 7. Zajišťující nalezení vnitřních bodů z množiny poskytnutých nově vytvořených bodů se dá v obecné rovině popsat jako:

1. Na základě poskytnutého pole xy popisujícího hranice oblasti stanov x_{min} , x_{max} představující nejnižší a nejvyšší hodnotu souřadnice x , stejně tak stanov y_{min} a y_{max} . Pro všechna $i = 1, \dots, n$, kde n je počet řádků NB , popisující souřadnice nových bodů v malém okolí hranice oblasti:
 - a) Ze souřadnic popisující souřadnici vyšetřovaného nového bodu $xy_{-}(i)$ veď 4 přímky
$$\begin{aligned} p_1 &= (x_{min}, x_{-}(i)), & p_2 &= (x_{-}(i), x_{max}), \\ p_3 &= (y_{min}, y_{-}(i)), & p_4 &= (y_{-}(i), y_{max}). \end{aligned} \quad (148)$$
 - b) Pomocí metody `intersections` hledej souřadnice průsečíků x_u, x_d, x_l, x_r mezi p_1, p_2, p_3, p_4 a xy , ty reprezentuj jako pole buněk.
 - c) Jsou-li současně všechna pole buněk x_u, x_d, x_l, x_r neprázdná (tedy průsečík existuje) potom daný vyšetřovaný nový bod $NB(i)$ označ jako vnitřní bod $VB(i)$, viz obr. 2.43 – A.
2. Označ jako výstup prvních k vnitřních bodů VB , kde k je nastaveno na 10.

Algoritmus 8. Zajišťující reprezentaci oblastí z hlediska jejich vzájemného topologického uspořádání, pracuje následovně:

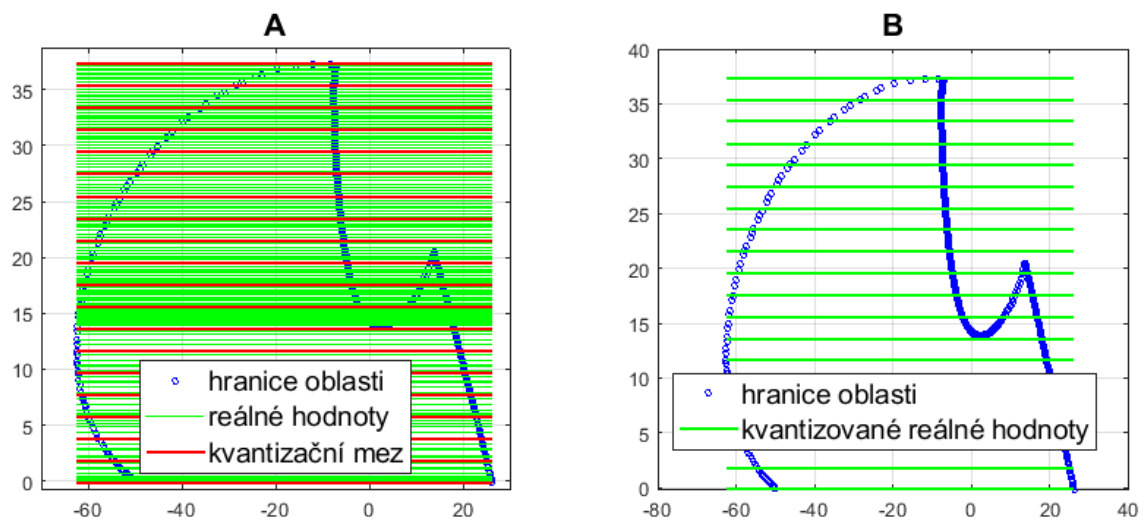
1. Stanov n_r jako počet poskytnutých oblastí R , sestav tabulku $T(n_r, n_r)$ o nulových hodnotách. Pro všechna $i = 1, \dots, n_r$:
 - a) Urči r_{ni} počet VB oblasti $R(i)$.
 - b) Pro všechna $j = 1, \dots, n_r$:
 - i. Pokud platí $i \neq j$, tak pomocí metody `F_ziskejVB` urči, jestli jsou vnitřní body oblasti $R(i)$ uvnitř oblasti vymezené hranicí $R_{xy}(j)$ (jinými slovy, zdali se $R(i)$ nachází uvnitř $R(j)$), urči r_{nj} jako počet nalezených vnitřních bodů.
 - ii. Pokud platí $r_{ni} = r_{nj}$, tak zaznamenej $T(j, i) = 1$.

2. Pro všechna $i = 1, \dots, n_r$ a $j = 1, \dots, n_r$ – pokud $T(i, j) \neq 0$, tak pro všechna $k = 1, \dots, n_r$ – v případě, že platí $T(j, k) \neq 0$, tak $T(i, k) = 0$.
3. Pro všechna $i = 1, \dots, n_r$:
 - a) Vytvoř nové pole xy_n , které bude reprezentovat hranice oblasti $R(i)$.
 - b) Pokud je i -tý řádek T nulový, tak ukonči danou iteraci i . V opačném případě pro všechna $k = 1, \dots, n_r$:
 - i. Pokud $T(i, k) \neq 0$ tak nalezní index m v poli xy_n , pro který má xy_n stejnou souřadnici jako začátek pole xy_k , k -té oblasti.
 - ii. Pokud index m existuje, tak aktualizuj pole xy_n jako $xy_n = xy_n(1, \dots, m), xy_k, xy_n(m, \dots, l)$ (kde l představuje délku pole xy_n), viz obr. 2.48 – B. V opačném případě přidej xy_k na konec pole xy_n .
 - c) Vytvoř novou oblast $R(i)$, jejíž hranici bude tvořit pole xy_n .

2.2.8 Vyplnění oblasti

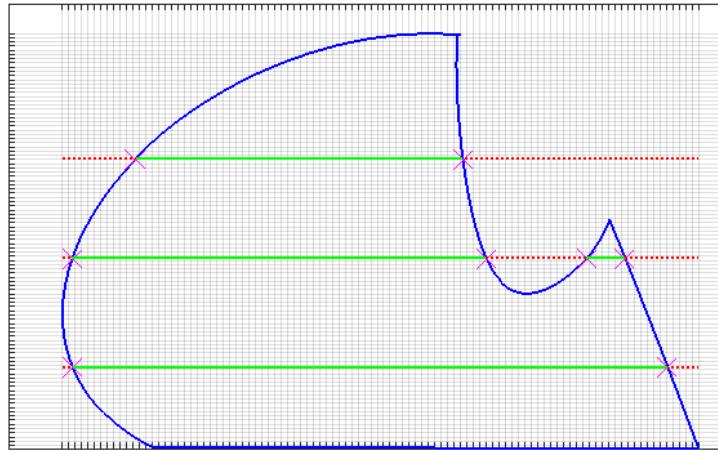
V této části je řešena otázka *vyplnění vnitřku oblasti*, resp. nalezení bodů, které se nachází v oblasti uzavřené segmenty křivky – uvnitř hranice oblasti. Vnitřek oblastí má zásadní význam z hlediska jejich reprezentace.

Princip funkce *nalezení bodů uvnitř oblasti* (lze chápat jako její vyplnění) spočívá ve vytvoření mřížky v ose x a y . Vytvořit mřížku v x -ových hodnotách není problém, v ose y by se však jednalo o nutnost vytvářet nový bod v místě rovnoměrného rozestupu osy x a naopak. Tuto situaci lze řešit tak, že je nejprve zajištěna dostatečná redundance hodnot (hladin y), ze kterých jsou následně vybrány ty, které nejvíce odpovídají rovnoměrným rozestupům (kvantizační mez, viz obr. 2.49 – A) mřížky v ose y . Takto vybrané hodnoty y jsou znázorněny na obr. 2.49 – B jako vodorovné úsečky.

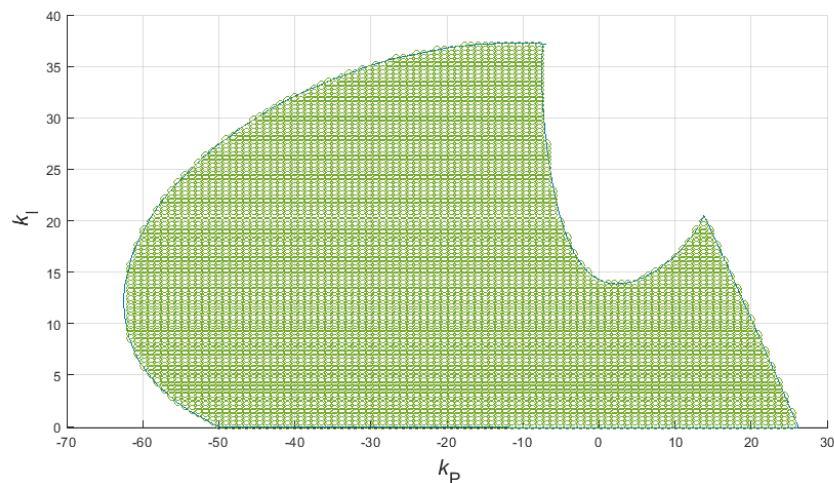


Obr. 2.49 – Princip konstrukce y -ových hodnot mřížky

Následně je z každé y -ové hodnoty (která popisuje hranici oblasti a odpovídá dané úrovni mřížky) vedena přímka v celém rozsahu rámu oblasti a průsečíky této přímky s hranicí oblasti x_0 jsou zaznamenány viz obr. 2.50. Je-li velikost pole x_0 sudé číslo, tak sousední liché dvojice hodnot v něm obsažených odpovídají vnitřku oblasti. Rám oblasti představují největší a nejmenší hodnoty v ose x a y , které tvoří obdélník.



Obr. 2.50 – Princip hledání bodů mřížky, které jsou uvnitř oblasti



Obr. 2.51 – Ukázka vyplněné oblasti

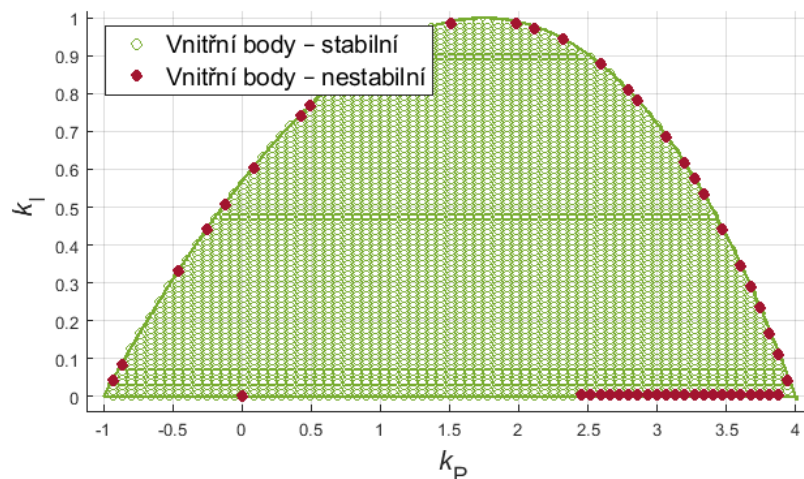
Algoritmus 9. Zajišťující nalezení bodů uvnitř oblasti, které náleží mřížce s hranicí rámu oblasti funguje následovně:

1. Vytvoř rám oblasti stanovením nejmenší a největší hodnoty v ose x a y , které popisují hranici oblasti, ty označ jako x_{min} , x_{max} , y_{min} , y_{max} .
2. Pomocí struktury try učin pokus o interpolaci xy v ose y (v některých případech může selhat),
3. Definuj O jako pole buněk dvojic průsečíků dané kvantizační hladiny s hranicí vyšetřované oblasti, $grxy$ jako prázdnou matici o dvou sloupcích a s_x jako pole délek všech mezilehlých hodnot x ,
4. Stanov hodnoty mřížky x_r v ose x jako r_n hodnot v intervalu $x_r \in \langle x_{min}, x_{max} \rangle$ a hodnoty mřížky y_r v ose y jako r_n hodnot v intervalu $y_r \in \langle y_{min}, y_{max} \rangle$.
5. Pro všechna $i = 1, \dots, r_n$:
 - a) Urči $z(i)$ jako index hodnoty hranice oblasti v ose y , která je nejbližší i -té hodnotě kvantizační meze (tedy $y_r(i)$) jako $z(i) = \operatorname{argmin}\{|y_r(i) - y|\}$.
6. Stanov koncové body přímky $x_l = [x_{min}, x_{max}]$,
7. Pro všechna $i = 1, \dots, L_y$, kde L_y představuje délku pole xy :
 - a) Stanov koncové body přímky $y_l = [y_{min}, y_{max}]$,
 - b) Nalezni všechny průsečíky v ose x mezi xy a xy_l ty označ jako pole x_0 ,
 - c) V případě, že $x_0 \geq 2$, tak pro všechna $j = 1, \dots, L_{x_0}$, kde $L_{x_0} - 1$ je délka pole x_0 , zjisti vzdálenost $d = |(x_0(j) - x_0(j+1))|$ a v případě, že $d < \frac{\max(s_x)}{100}$, tak polož $x_0(j+1) = x_0(j)$ (saturace).
 - d) Je-li pole souřadnic průsečíků x_0 neprázdné tak označ i -tý řádek O jako x_0 .
8. Pro všechna $i = 1, \dots, r_n$ stanov délku L pole x_0 a pokud platí $r \bmod n = 2$, tak pro všechna $j = 2, 4, 6, \dots, L$:
 - a) Polož x_1 jako $(j-1)$ -tý záznam v i -tém řádku pole O a x_2 jako j -tý záznam i -tého řádku pole O ,
 - b) Označ x_{in} jako indexy hodnot mřížky x_r , pro které platí $x_r \in \langle x_1, x_2 \rangle$.
 - c) Na konec matice $grxy$ přidej jako sloupce pole $x_r(x_{in})$ a $y(z(i))$.

2.2.9 Nalezení stabilních oblastí

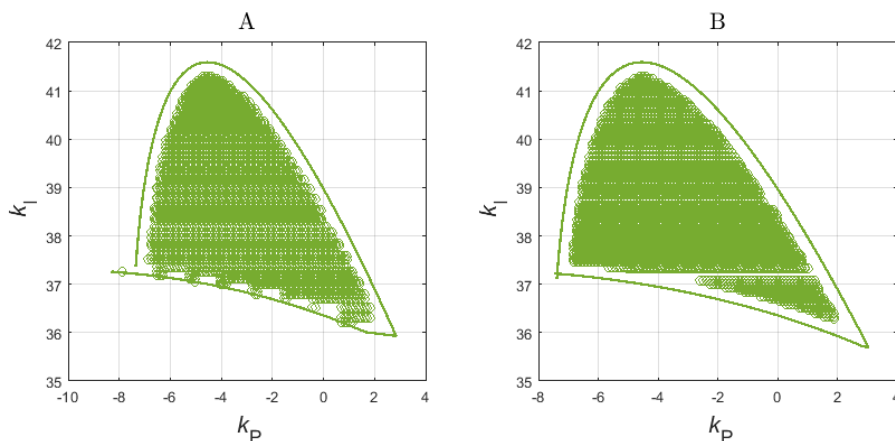
Tato kapitola se zabývá určením stabilních oblastí z množiny všech individuálně reprezentovaných oblastí. Z principu metody vyplývá, že pro ověření stability oblasti stačí vybrat jeden vnitřní bod a na něj následně aplikovat některé z kritérií ověření stability (viz oddíl 1.6).

Vlivem numerických nepřesností však může nastat situace, kdy většina bodů uvnitř oblasti je buď stabilní či nestabilní, ale některé body v okolí HK jsou označeny naopak (např. viz obr. 2.52). Aby nenastala situace, že jako testovací bod bude vybrán bod ležící na hranici nebo v její bezprostřední blízkosti, tak jsou body v okolí hranice oblasti odstraněny. Takto vynechané body nejsou uvažovány ani při optimalizaci, protože je zřejmé, že optimální nastavení v malém okolí hranice oblasti ležet nebude.



Obr. 2.52 – Nestabilní body na hranici stabilní oblasti

Odstranění bodů v okolí hranice oblasti vyžaduje, aby hranice měla dostatečné rozlišení, v opačném případě může nastat situace, která je znázorněna na obr. 2.53 – A, kde ve spodní části oblasti nejsou odstraněny všechny vnitřní body v daném okolí hranice. Aby k této situaci nedošlo, tak je potřeba hranici oblasti proložit dodatečnými body, a tím zvýšit její rozlišení, potom budou body v okolí hranice odstraněny korektně, viz obr. 2.53 – B.

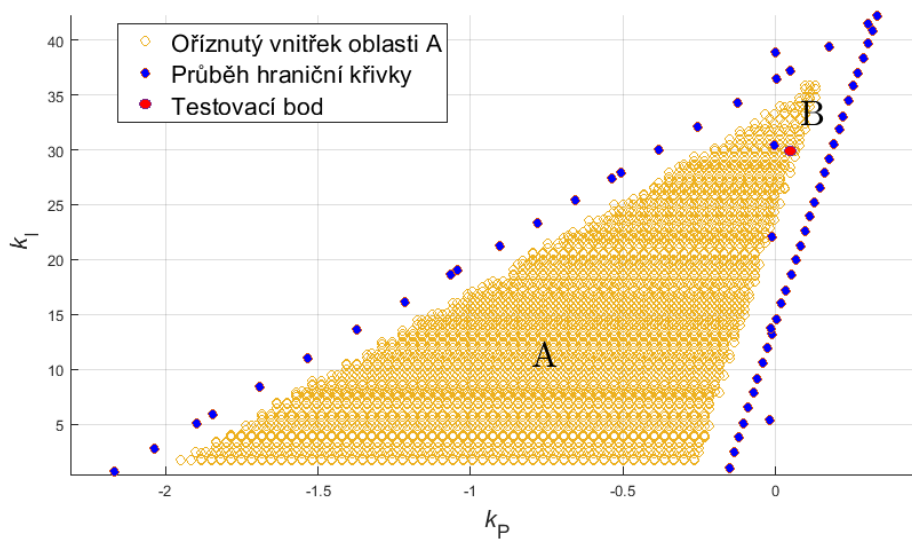


Obr. 2.53 – Nesprávné (A, vlivem rozlišení) a správné (B) odstranění vnitřních bodů v určité vzdálenosti podél hranice oblasti

Prakticky však může nastat situace, kde 2 oblasti nejsou vlivem rozlišení hraniční křivky navzájem oddělené. Tato situace nastává při kolapsu oblastí zařazením parametru derivační složky (viz dále) – v určité fázi zmenšování oblastí nastane situace, kdy je segment

HK představující hranici oblasti tvořen nedostatečným počtem bodů, a hranice není na úrovni algoritmu detekována. Tento jev je z podstaty nevyhnutelný, jelikož průběh HK není spojitý, ale jedná se o posloupnost diskrétních hodnot. Šance, že tento jev nastane v rovině (k_P, k_I) je malá, nicméně z podstaty může taková situace nastat také. Tento problém je tedy řešen již na úrovni této části algoritmu.

Uvažujme situaci na obr. 2.54 – je dána oblast A, která obsahuje stabilní podoblast B, tato podoblast však nebyla na úrovni algoritmu oddělena. Z množiny oříznutých vnitřních bodů oblasti A může být vybrán testovací bod, který náleží do stabilní neoddělené oblasti B. Na základě stability tohoto vybraného testovacího bodu by pak byla celá oblast A označena jako stabilní. Aby se pravděpodobnost vzniku této situace omezila, tak je náhodně zvoleno n testovacích bodů, přičemž aby byla oblast stabilní, tak musí platit, že je stabilních i všech n testovacích bodů.



Obr. 2.54 – Chybné označení oblasti vlivem nepřesnosti

2.3 VYUŽITÍ OBLASTI STABILITY PRO OPTIMÁLNÍ NÁVRH PI REGULÁTORU

Jakmile je získána stabilní oblast, tak je možné v rámci jejích vnitřních bodů hledat bod takový, který bude odpovídat optimálnímu nastavení regulátoru. V této práci je pro všechny vnitřní body oblasti počítáno integrální kritérium. Hodnota integrálního kritéria (jmenovitě kvadratické regulační plochy a její modifikace) se počítá postupem uvedeným v pododdíle 1.8.1. Přičemž jako optimální nastavení je považována taková

sada nastavitelných parametrů PI regulátoru, pro kterou je hodnota integrálního kritéria nejnižší.

Uvažujme následující sadu 4 stabilních (A, B, C, D) a 2 nestabilních testovacích systémů (E, F):

$$\begin{aligned}
 G_A(s) &= \frac{6}{s^3 + 6s^2 + 11s + 6}, \\
 G_B(s) &= \frac{s^3 + 200s^2 - 89s + 11}{s^4 + 4s^3 + 6s^2 + 4s + 1}, \\
 G_C(s) &= \frac{-34,16s^3 - 144,4s^2 + 7047s + 557,2}{s^5 + 13,18s^4 + 95,93s^3 + 14,61s^2 + 31,94s + 0,1}, \\
 G_D(s) &= \frac{-s^3 - 200s^2 + 120s + 15}{0,08s^4 + 1,24s^3 + 3,237s^2 + 3,037s + 0,96}, \\
 G_E(s) &= \frac{s - 1}{s^2 + 0,8s - 0,2}, \\
 G_F(s) &= \frac{10s^2 - 50s + 4}{s^5 + 2s^4 + 32s^3 + 14s^2 - 4s + 50}.
 \end{aligned} \tag{149}$$

Pro systémy (149) byla způsobem uvedeným v předchozích pododdílech získána oblast stability a následně v ní byly pomocí optimalizace nalezeny body, které odpovídají optimálnímu nastavení. Jako kritérium byla použita kvadratická regulační plocha (73) a její modifikace (82), která zahrnuje derivaci regulované veličiny násobenou váhovým koeficientem κ . Kvadratická regulační plocha bez uvažování modifikace (73) odpovídá rozšířenému kritériu (82) pro $\kappa = 0$.

Testovací systém (149) – A představuje stabilní nekmitavý systém 3. řádu. Na obr. 2.55 je vidět, že regulační pochod pro $\kappa = 0$ je příliš kmitavý. Volbou vyšší hodnoty κ je však možné získat aperiodický regulační pochod, aniž by byl významně prodloužen regulační děj.

Testovací systém (149) – B představuje stabilní a výrazně neminimálně-fázový systém s komplexně sdruženou nulou v pravé polorovině komplexní roviny. Na obr. 2.56 je vidět regulační pochod pro různé hodnoty koeficientu κ . Již regulační pochod pro $\kappa = 0$ představuje relativně uspokojivý výsledek bez překmitu. Zvýšení hodnoty κ vede na prodloužení regulačního děje a snížení míry kmitavosti. Pozorovatelný je slabý překmit pro $\kappa = 5$.

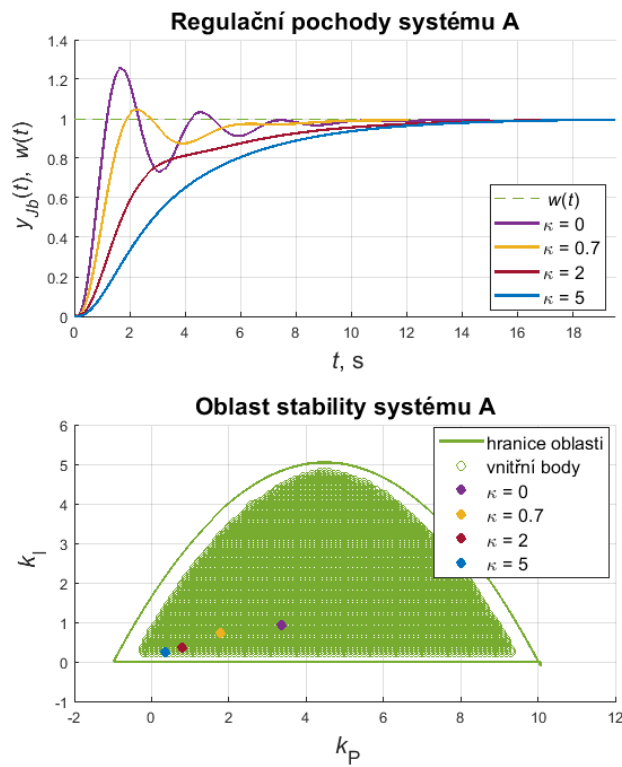
Testovací systém (149) – C představuje stabilní systém s výraznou dynamikou a jednou nestabilní nulou. Na obr. 2.57 je vidět, že regulační pochod základního kritéria kvadratické regulační plochy (73) je až příliš kmitavý. Také je však vidět vliv koeficientu

κ na snížení míry kmitavosti. Ani zvýšení koeficientu κ by, v případě tohoto systému, nevedlo na odstranění vyššího překmitu. Nově nalezené body, odpovídající optimálním nastavitelným parametrům regulátoru, mají totiž tendenci směřovat k okraji oblasti stability pro zvyšování hodnoty κ . Oblast stability tohoto systému patří k nejmenším nalezeným.

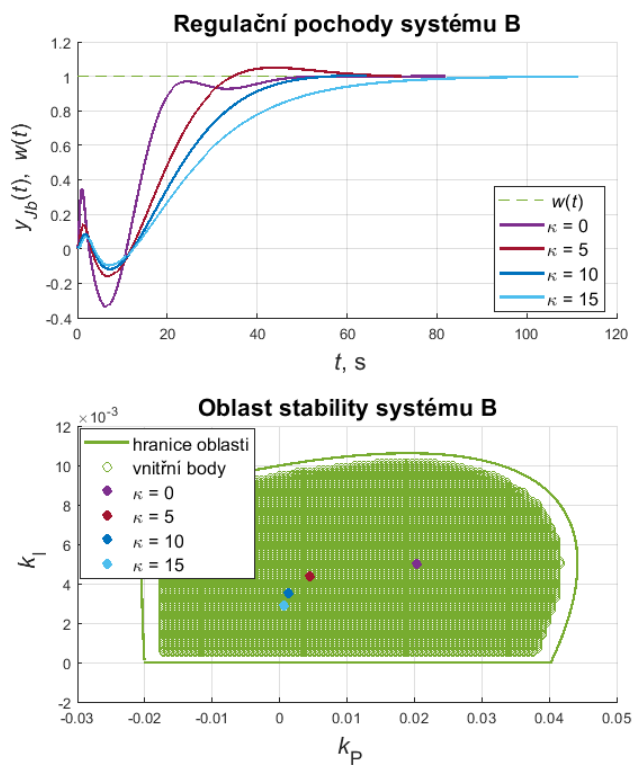
Testovací systém (149) – D představuje stabilní systém s jednou nestabilní nulou. Na obr. 2.58 je vidět vliv koeficientu κ na míru kmitavosti, současně se však prodlužuje regulační pochod. Již nastavení získané pro $\kappa = 0$ by se dalo považovat za vhodné.

Testovací systém (149) – E představuje nestabilní systém s jedním nestabilním pólem. Na obr. 2.59 je vidět vliv koeficientu κ na míru kmitavosti, současně se však prodlužuje regulační pochod. Ani pro vyšší hodnoty koeficientu κ nedochází ke snížení velikosti relativního překmitu. Již samotnou existenci oblasti stability a její nalezení lze však považovat za významnou výhodu metody hraniční křivky.

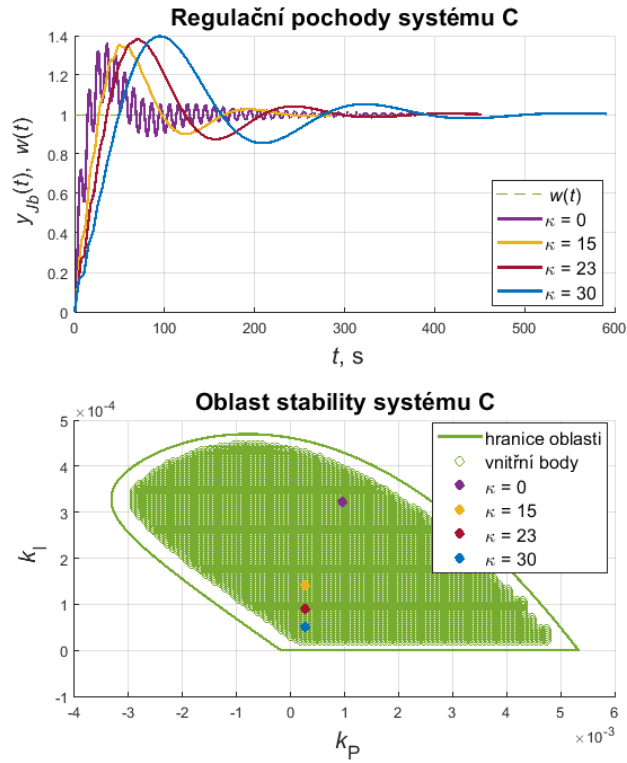
Testovací systém (149) – F představuje nestabilní systém s dvěma nestabilními nulami a komplexně sdruženým nestabilním pólem. Na obr. 2.60 je vidět vliv koeficientu κ na míru kmitavosti. V tomto případě nemá zvýšení hodnoty κ příliš velký vliv na regulační pochod, který je velmi kmitavý. Stejně jako v předchozím případě je však nalezení oblasti stabilizujících parametrů výhodné.



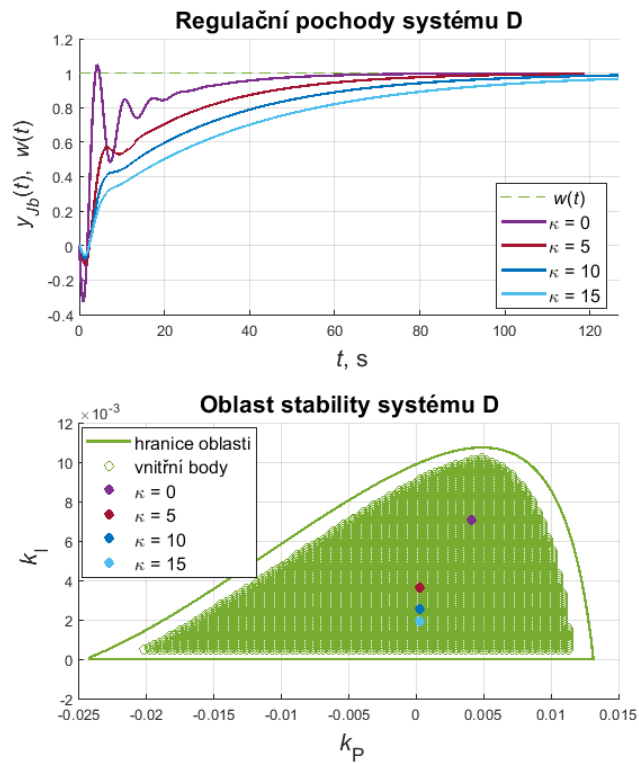
Obr. 2.55 – Regulační pochody a oblast stability testovacího systému (149) – A



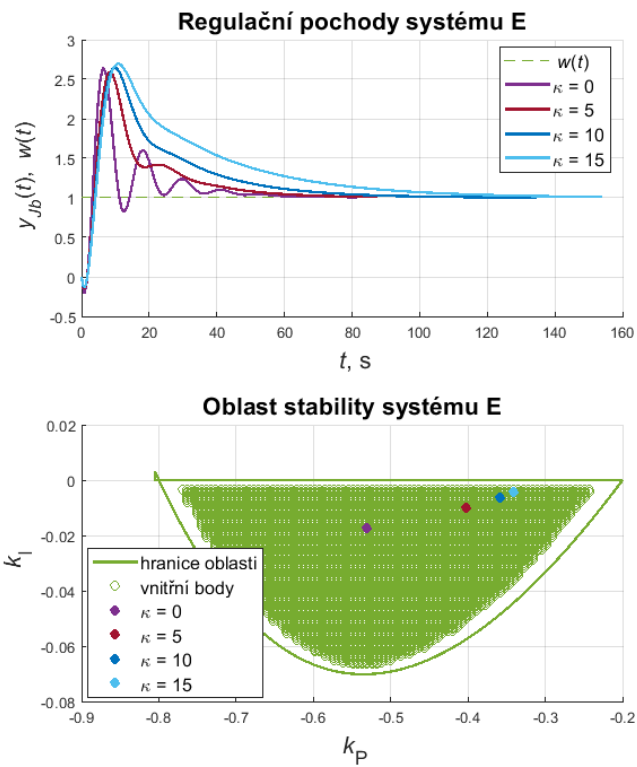
Obr. 2.56 – Regulační pochody a oblast stability testovacího systému (149) – B



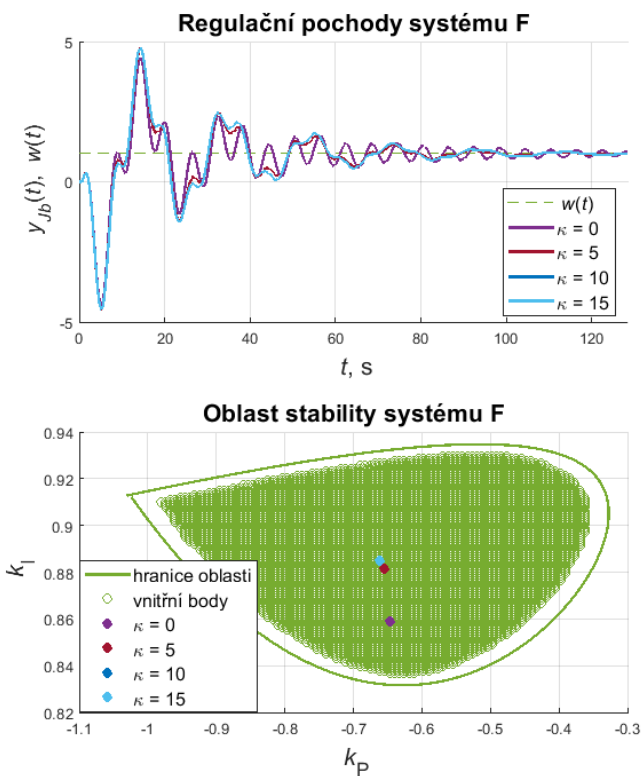
Obr. 2.57 – Regulační pochody a oblast stability testovacího systému (149) – C



Obr. 2.58 – Regulační pochody a oblast stability testovacího systému (149) – D



Obr. 2.59 – Regulační pochody a oblast stability testovacího systému (149) – E



Obr. 2.60 – Regulační pochody a oblast stability testovacího systému (149) – F

2.4 ROZŠÍŘENÍ O PARAMETR DERIVAČNÍ SLOŽKY PID REGULÁTORU

V tomto oddíle je popsáno rozšíření výpočtu o parametr derivační složky regulátoru k_D . Jedná se o jakousi nadstavbu předchozího přístupu, který spočívá ve výpočtu a vykreslení hraniční křivky v rovině (k_P, k_I) . Uvažováním parametru k_D lze získat průběh HK v prostoru parametrů (k_P, k_I, k_D) .

Postupem uvedeným v oddílu 2.1 lze získat průběh hraniční křivky v rovině (k_P, k_I) pro fixní hodnotu k_D . To vyžaduje rozšíření původního vztahu (101), označme

$$k_I = k_I + k_D \omega^2, \quad (150)$$

následným dosazením do (108) lze získat (Tan et al., 2006)

$$k_I = \frac{YQ - XY}{QU - RS} + k_D \omega^2 = \frac{YQ - XS + k_D(QU - RS)\omega^2}{QU - RS}, \quad (151)$$

přičemž k_P zůstává stejné jako ve vztahu (108) tedy (Tan et al., 2006)

$$k_P = \frac{X(\omega)U(\omega) - Y(\omega)R(\omega)}{Q(\omega)U(\omega) - R(\omega)S(\omega)}. \quad (152)$$

Příklad – uvažujme systém popsany přenosem (Tan et al., 2006)

$$G(s) = \frac{N(s)}{D(s)} = \frac{s^3 - 4s^2 + s + 2}{s^5 + 8s^4 + 32s^3 + 46s^2 + 46s + 17}, \quad (153)$$

jeho frekvenční přenos odpovídá

$$G(i\omega) = \frac{N(i\omega)}{D(i\omega)} = \frac{-i\omega^3 + 4\omega^2 + i\omega + 2}{i\omega^5 + 8\omega^4 - 32i\omega^3 - 46\omega^2 + 46i\omega + 17}. \quad (154)$$

Čitatel a jmenovatel frekvenčního přenosu lze rozložit na

$$N = N_e + i\omega N_o; D = D_e + i\omega D_o, \quad (155)$$

tedy

$$N_e = 4\omega^2 + 2; N_o = -\omega^2 + 1; D_e = 8\omega^4 - 46\omega^2 + 17; D_o = \omega^4 - 32\omega^2 + 46. \quad (156)$$

Následným dosazením do zavedených substitucí

$$\begin{aligned}
Q &= -\omega^2 N_o = (-\omega^2)(-\omega^2 + 1), \\
S &= \omega N_e = \omega(4\omega^2 + 2), \\
X &= \omega^2 D_o = \omega^2(\omega^4 - 32\omega^2 + 46), \\
R &= N_e = 4\omega^2 + 2, \\
U &= \omega N_o = \omega(-\omega^2 + 1), \\
Y &= -\omega D_e = -\omega(8\omega^4 - 46\omega^2 + 17),
\end{aligned} \tag{157}$$

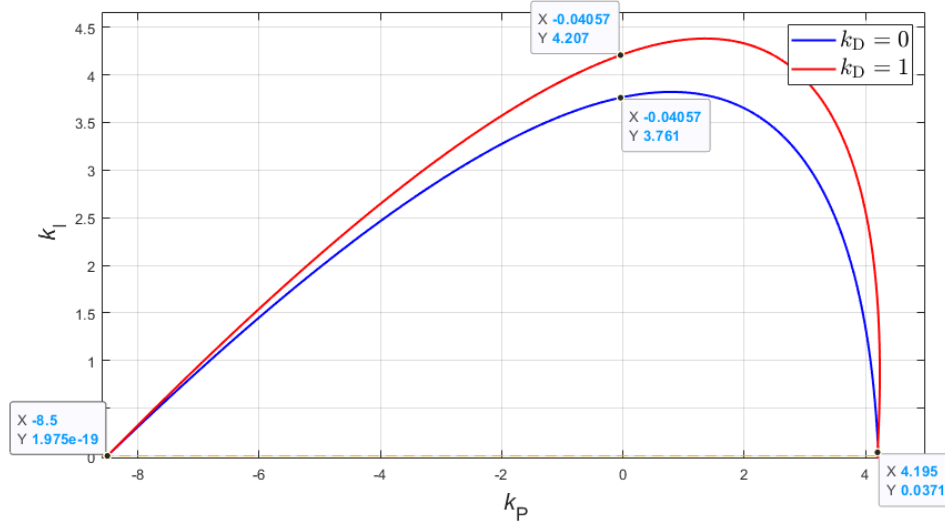
jak již bylo zmíněno, k_P na parametru derivační složky nezáleží

$$k_P = \frac{\omega^8 - 65\omega^6 + 246\omega^4 - 22\omega^2 - 34}{\omega^6 + 14\omega^4 + 17\omega^2 + 4}, \tag{158}$$

a dosazením do odvozeného vztahu pro výpočet integrační složky (151)

$$\begin{aligned}
k_I &= \frac{YQ - XS + k_d \omega^2 (QU - RS)}{QU - RS} = \\
&= \frac{-\omega(8\omega^4 - 46\omega^2 + 17)(-\omega^2)(-\omega^2 + 1) - \omega^2(\omega^4 - 32\omega^2 + 46)\omega(4\omega^2 + 2)}{(-\omega^2)(-\omega^2 + 1)\omega(-\omega^2 + 1) - 4\omega^2 + 2\omega(4\omega^2 + 2)} + \\
&\quad + \frac{k_d \omega^2 ((-\omega^2)(-\omega^2 + 1)\omega(-\omega^2 + 1) - 4\omega^2 + 2\omega(4\omega^2 + 2))}{(-\omega^2)(-\omega^2 + 1)\omega(-\omega^2 + 1) - 4\omega^2 + 2\omega(4\omega^2 + 2)} = \\
&= \frac{(-8\omega^9 + 54\omega^7 - 63\omega^5 + 17\omega^3) - (4\omega^9 - 126\omega^7 + 120\omega^5 + 92\omega^3)}{(-\omega^8 + 2\omega^5 - \omega^3) - (16\omega^5 + 16\omega^3 + 4\omega)} + \\
&\quad + \frac{k_d \omega^2 ((-\omega^8 + 2\omega^5 - \omega^3) - (16\omega^5 + 16\omega^3 + 4\omega))}{(-\omega^8 + 2\omega^5 - \omega^3) - (16\omega^5 + 16\omega^3 + 4\omega)} = \\
&= \frac{-12\omega^9 + 180\omega^7 - 183\omega^5 - 75\omega^3 + k_d \omega^2 (-\omega^7 - 14\omega^5 - 17\omega^3 - 4\omega)}{-\omega^7 - 14\omega^5 - 17\omega^3 - 4\omega} = \\
&= \frac{12\omega^9 - 180\omega^7 + 183\omega^5 + 75\omega^3 + k_d \omega^9 + 14k_d \omega^7 + 17k_d \omega^5 + 4k_d \omega^3}{\omega^7 + 14\omega^5 + 17\omega^3 + 4\omega} = \\
&= \frac{\omega^9(12 + k_d) - \omega^7(180 - 14k_d) + \omega^5(183 + 17k_d) + \omega^3(75 + 4k_d)}{\omega^7 + 14\omega^5 + 17\omega^3 + 4\omega} = \\
&= \frac{\omega(\omega^8 a_1 - \omega^6 a_2 + \omega^4 a_3 + \omega^2 a_4)}{\omega(\omega^6 + 14\omega^4 + 17\omega^2 + 4)} = \frac{a_1 \omega^8 - a_2 \omega^6 + a_3 \omega^4 + a_4 \omega^2}{\omega^6 + 14\omega^4 + 17\omega^2 + 4},
\end{aligned} \tag{159}$$

je dán výsledek, kde $a_1 = 12 + k_d$; $a_2 = 180 - 14k_d$; $a_3 = 183 + 17k_d$; $a_4 = 75 + 4k_d$. Na obr. 2.61 lze vidět průběh $l(\omega)$ pro hodnoty parametru $k_D = 0$ a $k_D = 1$, přičemž výsledek souhlasí s literaturou (Tan et al., 2006).



Obr. 2.61 – Vliv derivační složky na deformaci průběhu HK (Tan et al., 2006)

V rámci nalezení ukončovací frekvence je hledán průsečík otevřeného regulačního obvodu s $\mathcal{J}[G(i\omega)] = 0$

$$G_o(s) = \frac{k_P(1 + \frac{1}{T_I} + T_D s)G(s)}{1 + k_P(1 + \frac{1}{T_I} + T_D s)G(s)}, \quad (160)$$

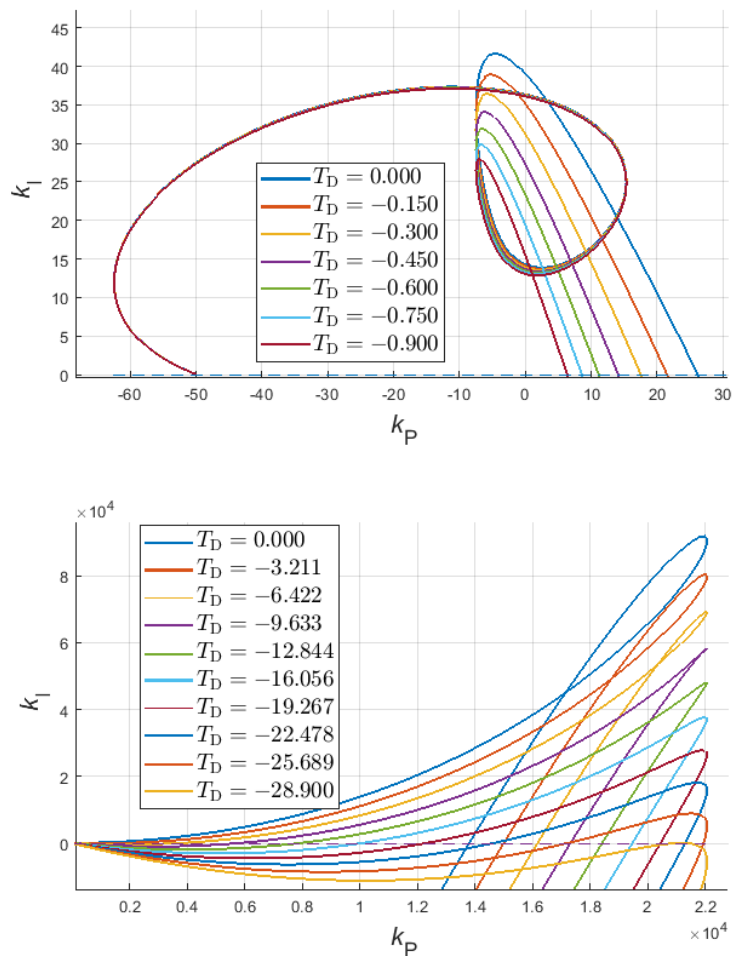
průsečík s $\mathcal{J}[G(i\omega)] = 0$ sice na hodnotě integrační složky závisí, je však hledán takový průsečík, pro který je integrační složka nulová (je hledán průsečík průběhu HK s osou $k_I = 0$). Proto lze uvažovat

$$G_o(s) = \frac{k_P(1 + T_D s)G(s)}{1 + k_P(1 + T_D s)G(s)}. \quad (161)$$

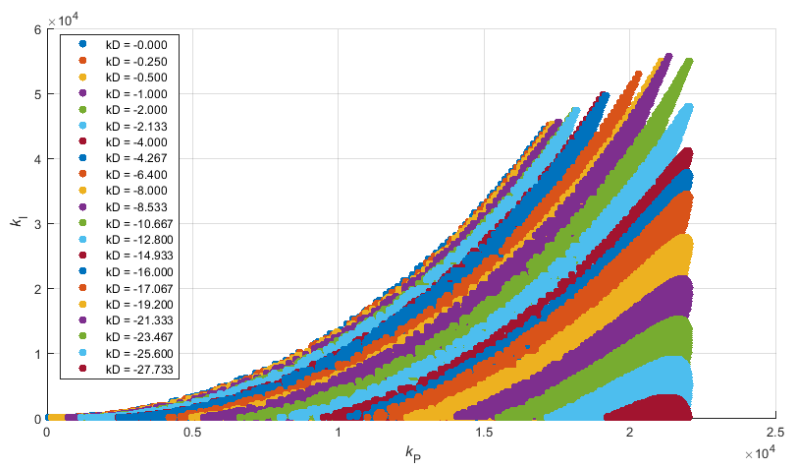
Ekvivalentně je potřeba upravit i vzorce pro stanovení průběhu $l(\omega)$, tzn. upravit vztah (151) na

$$k_I = \frac{YQ - XY}{QU - RS} + k_D \omega^2 = \frac{YQ - XS + k_P(QU - RS)\omega^2}{QU - RS}. \quad (162)$$

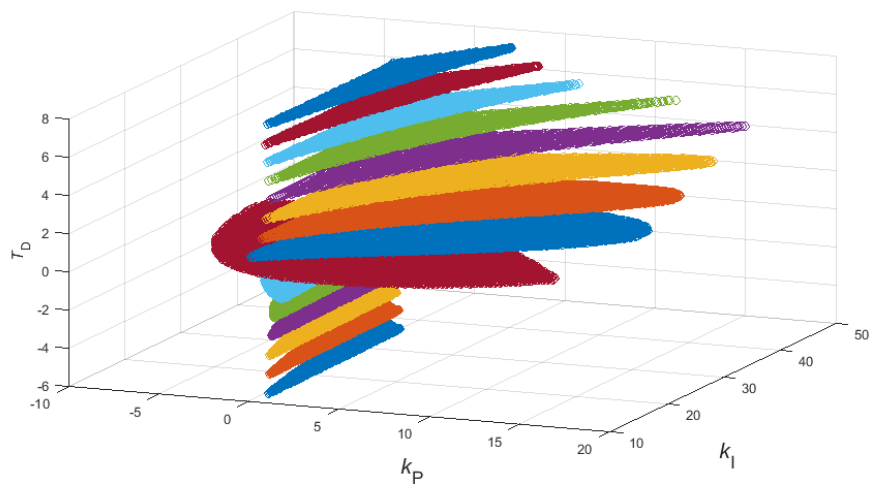
Na základě postupu uvedeném v oddílu 2.2 lze následně získat oblasti stability pro specifikované hodnoty T_D . Deformaci oblasti stability pro různé systémy a různé hodnoty parametru derivační složky zachycuje obr. 2.62, 2.63, 2.64, 2.65 a obr. 2.66.



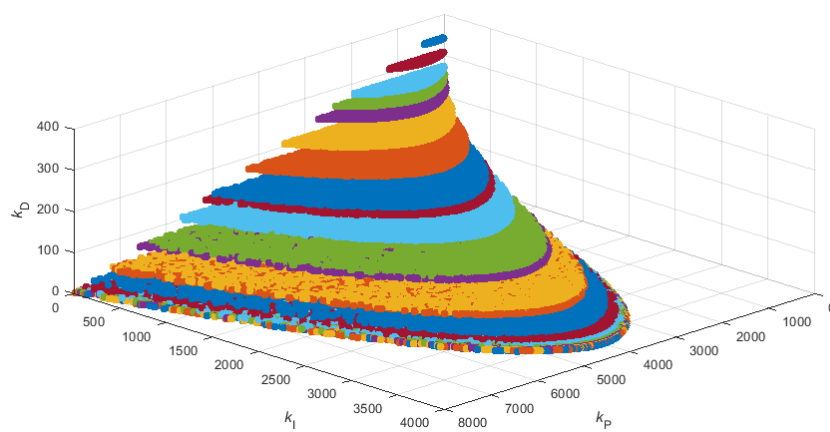
Obr. 2.62 – Příklad deformace průběhů HK pro různé hodnoty parametru T_D , jehož důsledkem je kolaps oblastí stability



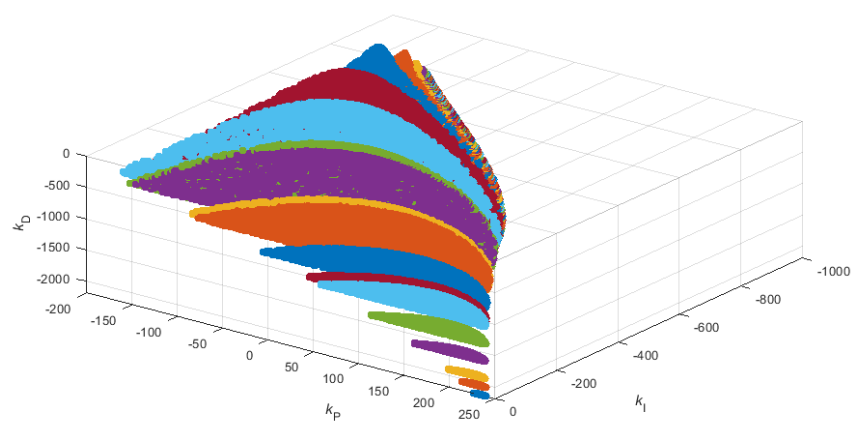
Obr. 2.63 – Příklad deformace oblasti stability pro různé hodnoty parametru k_D



Obr. 2.64 – Příklad deformace oblasti stability pro různé hodnoty parametru T_D

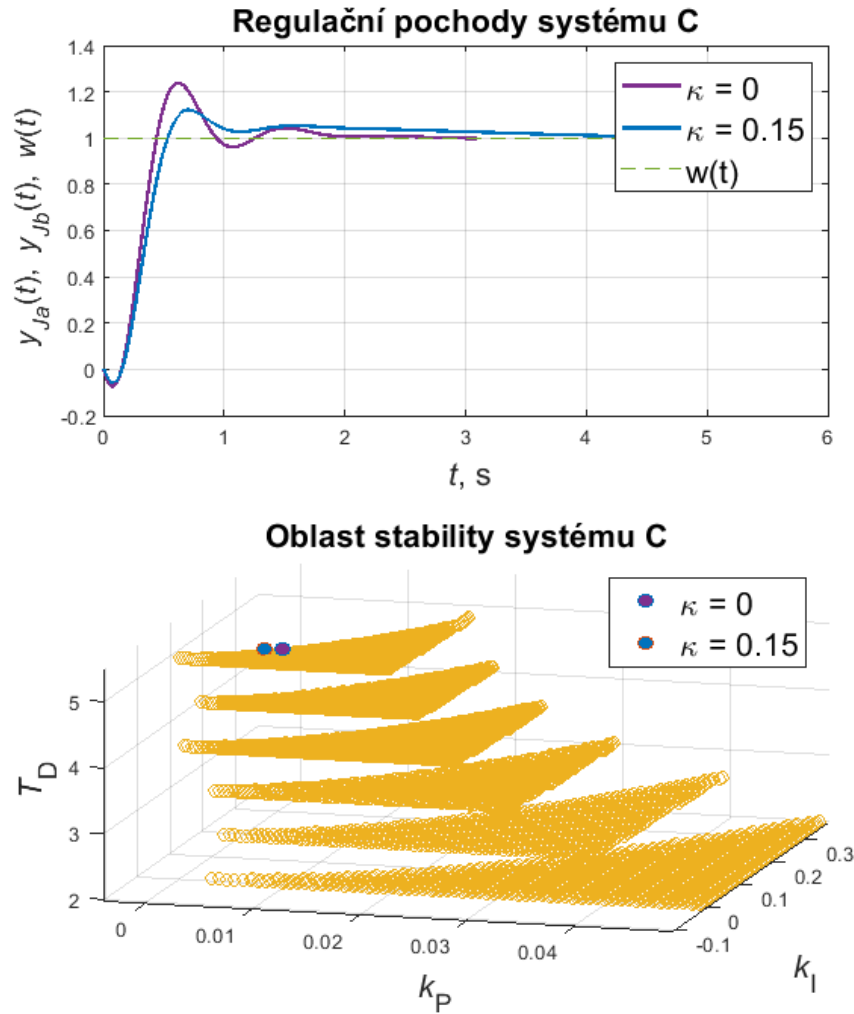


Obr. 2.65 – Příklad deformace oblasti stability pro různé hodnoty parametru T_D



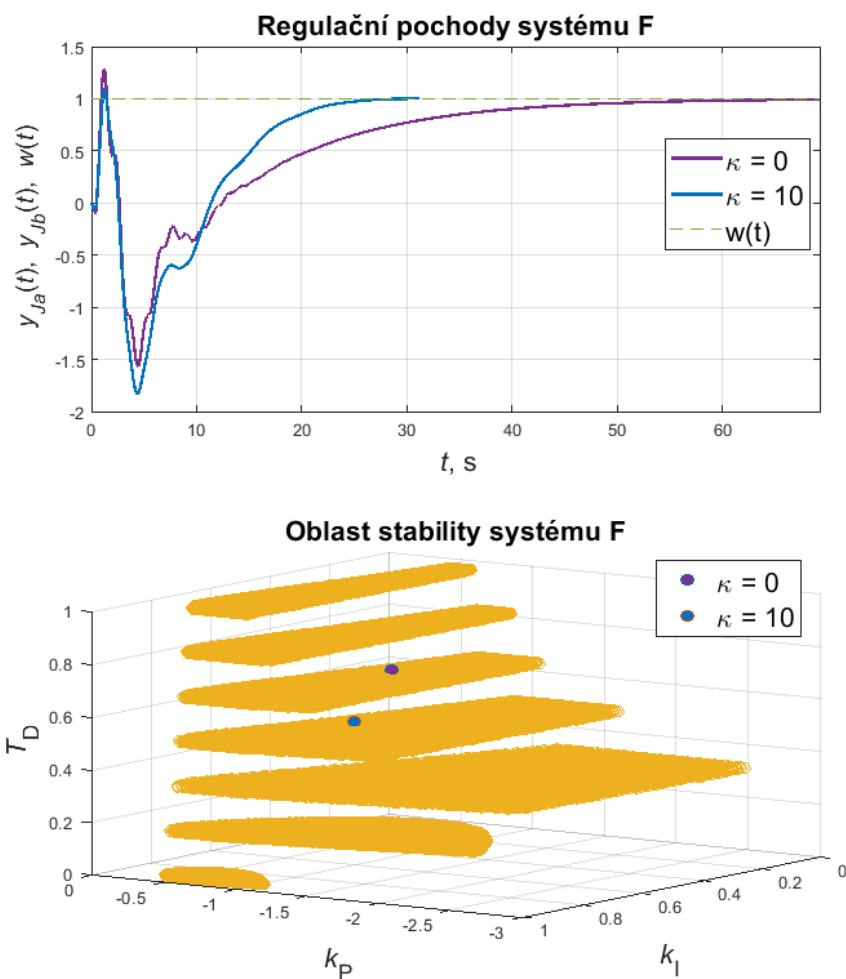
Obr. 2.66 – Příklad deformace oblasti stability pro různé hodnoty parametru T_D

Stejně jako v rovině (k_P, k_I) , lze hledat *optimální nastavení* PID regulátoru i v rovině (k_P, k_I, k_D) . Uvažujme získané regulační pochody systému (149) – C a systému (149) – F. Oproti původním RP (viz obr. 2.57, 2.60) je vidět, že zařazením parametru derivační složky T_D došlo k významnému zlepšení regulačního pochodu, viz obr. 2.67, 2.68.



Obr. 2.67 – Regulační pochody systému (149) – C odpovídající optimálním nastavitelným parametrům PID regulátoru v prostoru parametrů (k_P, k_I, k_D)

Jelikož nejnižší a nejvyšší hodnoty derivační složky, pro kterou by ještě existovala stabilní oblast, nejsou dopředu známé, tak je tato volba ponechána na uživateli. Alternativně by program bylo možné rozšířit ve smyslu doplnění algoritmu, který by např. metodou půlení intervalu našel hodnoty T_{Dmin} a T_{Dmax} , a stanovil n rovnoměrně rozložených hodnot mezi nimi.



Obr. 2.68 – Regulační pochody nestabilního systému (149) – F odpovídající optimálním nastavitelným parametrům PID regulátoru v prostoru parametrů (k_P, k_I, k_D)

Rozšířením uvažované dimenze se také zvyšuje šance, že program skončí chybou. Jedná se zejména o situace, kdy je oblast pro určitou hodnotu derivační složky příliš malá (viz např. obr. 2.54). Celková robustnost kódu by se dala zvýšit hledáním stabilních bodů nikoli stabilních oblastí. To by znamenalo testovat stabilitu pro každý bod v každé oblasti a algoritmus ukončit až v případě, že žádný bod není stabilní. To je však v rozporu z principem metody.

Rozšíření programu o parametr T_D lze popsat následnými obecnými kroky:

1. Zvol počáteční (T_A) a koncovou (T_B) hodnotu derivační složky a parametr n . Urči pole T_D , které představuje n rovnoměrně rozložených hodnot mezi T_A a T_B .

2. Pro všechny $i = 1, \dots, n$:
 - a) Urči stabilní oblasti pro $T_D(i)$ a ty ulož do pole $R(i)$, která na daném indexu i reprezentuje matici významných proměnných popisujících stabilní oblast.
 - b) Nastane-li situace, že není vrácena žádná stabilní oblast, tak program ukonči.
3. Pokud alespoň jeden seznam $R(i)$ není prázdný, tak vykresli jednotlivé řezy v prostoru parametrů (k_P, k_I, k_D) .
4. Pro všechny řezy v prostoru parametrů (k_P, k_I, k_D) nalezni takové nastavení, které má nejnižší hodnotu kritéria, tu zvýrazni a vykresli také odpovídající regulační pochody.

ZÁVĚR

V rámci této práce byla implementována jedna z moderních metod umožňujících získání oblasti stability pro zadaný lineární systém. Jedná se o metodu hraniční křivky (Tan et al., 2006) využívající tzv. D-dekompozice. Metoda spočívá ve výpočtu průběhu hraniční křivky, která svým průběhem dělí rovinu parametrů (k_P, k_I) na stabilní a nestabilní podoblasti. Získaná oblast stability představující celou množinu přípustných nastavení PI(D) regulátoru byla následně využita k návrhu PI(D) regulátoru pomocí integrálního kritéria.

Pro implementaci metody hraniční křivky bylo využito matematických nástrojů z oblastí, které s teorií řízení přímo nesouvisí, např. se jedná o nástroje z oblasti analytické geometrie či teorie grafů. Konkrétněji, analytická geometrie byla využita k nalezení významných průsečíků hraniční křivky (tedy průsečíků křivky se sebou samou a dále také průsečíků hraniční křivky s horizontální osou). Dále byla využita např. při výpočtu průběhu hraniční křivky či při hledání bodů v blízkosti hranic oblastí. Teorie grafů byla využita k reprezentaci průběhu hraniční křivky a to tak, že nalezené průsečíky představují vrcholy a hrany reprezentují segmenty křivky, které příslušné vrcholy spojují. Využití nástrojů teorie grafů také následně umožnilo jednotlivé oblasti od sebe oddělit, individuálně reprezentovat a následně graficky znázornit. Ukázalo se, že pro korektní reprezentaci resp. zobrazení oblastí bylo třeba provést jejich topografické uspořádání na základě vzájemné inkluze.

Předem je vhodné zdůraznit skutečnost, že podstatnou část řešení tvořilo ošetření potenciálních situací, které prakticky nastávaly velice zřídka. Například v publikovaném článku (Tan et al., 2006) byl představen systém, pro který hraniční křivka vymezení několik oblastí, to však nebývá typické a bylo potřeba získat více systémů vykazujících podobné chování. Z tohoto důvodu bylo v rámci implementace potřeba navrhnout také rozsáhlou sadu testovacích soustav ve formě obrazových přenosů, které vedly na netriviální průběh hraniční křivky. Tato sada testovacích soustav byla získána na základě rešerše odborné literatury a prostřednictvím experimentálního návrhu. Testovací systémy byly využity pro ověření správné funkčnosti algoritmu, zejména s ohledem na jeho robustnost a ověření správné konstrukce příslušných grafů. Některé průběhy hraniční křivky bylo také nutné navrhnout bez návaznosti na konkrétní systém popsany obrazovým přenosem.

V průběhu řešení bylo potřeba řešit různě složité, často i neočekávané, problémy. Příkladem komplikací, které bylo v rámci práce potřeba řešit, byl např. duplicitní průsečík v místě, kde křivka protíná samu sebe, přičemž správné nalezení průsečíků je pro implementaci zásadní. Významné bylo také řešení ukončení výpočtu, pro které bylo v práci navrženo několik postupů. Jako univerzální byl do finální verze zvolen ten, který spočívá ve výpočtu průběhu hraniční křivky až do okamžiku, kdy průběh hraniční dosáhne hrany ohraničujícího obdélníku. Pro samotný výpočet bodů popisujících průběh hraniční křivky byla implementována řada postupů s tím, že vypočítané hodnoty bylo následně potřeba rozšířit a vhodně upravit ve smyslu jejich interpolace. Komplikaci také představovala skutečnost, že mnoho funkcí pro práci s grafy programu MATLAB (verze 2020a) neumí pracovat s multigrafem a vyžadují na svém vstupu obyčejný graf, tedy graf neobsahující smyčky a vícenásobné hrany. Dílčí komplikací bylo např. také implicitní řazení hran v MATLAB reprezentaci grafu vzestupně podle vrcholů.

Nejsložitější situaci však představovala konstrukce kompletního grafu a zajištění topologického uspořádání oblastí. Na úrovni výpočtu bylo potřeba rozlišit mezi různými typy průsečíků a dále také rozlišit různé typy oblastí. Těch bylo nalezeno celkem sedm, k jejich reprezentaci nebylo možné přistoupit jednotně – reprezentace jednoho typu oblasti vedla na kombinatorickou úlohu, vyžadující zajištění správné orientace hran a následného výpočtu obsahu s cílem vybrat z možných kombinací tu nejmenší, jiný typ oblasti zase vedl na problém s překrýváním oblastí. Pro řešení dílčích úloh byla implementována řada metod, které se liší svojí efektivitou a svými vlastnostmi, přičemž do finální verze práce byly zařazeny ty nejvhodnější z nich.

Při implementaci bylo možné zavést několik zjednodušujících předpokladů, které by vedly na podstatně jednodušší řešení, takovým zjednodušením mohlo být např. ukončením výpočtu průběhu hraniční křivky v místě prvního průsečíku s horizontální osou $k_1 = 0$, či konstrukce samostatného grafu pro každou dvojici sousedních hodnot kritické frekvence. Při rozšíření práce o derivační složku bylo možné pro každý řez vracet stabilní body, nikoli stabilní oblasti, protože v místě kolapsu průběhu hraniční křivky začaly vymezené oblasti vykazovat anomální chování. Přičemž z principu je tato situace, vlivem diskrétní reprezentace hraniční křivky, nevyhnutelná. Uvedená zjednodušení však nebyla na základě rozhodnutí autora uvažována.

Je vhodné zmínit, že průběhy hraniční křivky mohou být velmi různorodé a i přes vynaloženou snahu vytvořit obecný algoritmus může v některých případech hledání stabilních oblastí selhat.

Univerzálnost metody hraniční křivky byla ověřena pro různé typy dynamických systémů, a to i těch nestabilních, které se někdy v průmyslové sféře vyskytují a některé jsou dokonce navrženy tak, aby vykazovaly nestabilní chování. Právě možnost získání oblasti stabilizujících parametrů patří mezi nejvýznamnější výhody této metody. Některé nalezené výjimky, pro které metoda či tato konkrétní implementace není vhodná, byly demonstrovány a diskutovány. Výsledky byly ověřeny analyticky a také experimentálně, a to jak ve smyslu porovnání výsledků s autory článku (Tan et al., 2006), tak ve smyslu simulace.

V průběhu implementace byly zjištěny také další zajímavé vlastnosti implementované metody. Konkrétněji se jedná např. o skutečnost, že nestačí výpočet hraniční křivky ukončit v posledním průsečíku s horizontální osou, a také to, že nestačí konstruovat zjednodušený obecný, ale kompletní planární graf.

POUŽITÁ LITERATURA

- BAJCINCA, Naim, 2006. Design of robust PID controllers using decoupling at singular frequencies. *Automatica*. **42**(11), 1943–1949. ISSN 0005-1098. DOI: <https://doi.org/10.1016/j.automatica.2006.06.006>.
- BALÁTĚ, Jaroslav, 2003. *Automatické Řízení*. 1. vyd. Praha: BEN – technická literatura. ISBN 80-7300-020-2.
- CVEJN, Jan, 2017. *Automatizace 1*. Pardubice: Přednášky předmětu.
- D'ERRICO, John, 2021. arclength. *MATLAB Central File Exchange* [online] [cit. 2021-04-12]. URL: <https://www.mathworks.com/matlabcentral/fileexchange/34871-arclength>.
- DUŠEK, František, 2019. *Spojité řízení*. Pardubice: Přednášky předmětu.
- FRANKLIN GENE F. Powell J. David, Emami-Naeini Abbas, 2015. *Feedback control of dynamic systems*. 7th edition. Boston: Pearson. ISBN 9780133496598.
- GAZDOŠ F., Kolařík J., 2013. Database Of Unstable Systems: A New Site For Models Of Unstable Processes. In: *ECMS 2013 Proceedings edited by: Webjorn Rekdalsbakken, Robin T. Bye, Houxiang Zhang*. ECMS, 5/2013. DOI: [10.7148/2013-0422](https://doi.org/10.7148/2013-0422).
- GOLNARAGHI M. F, Kuo Benjamin C., 2017. *Automatic Control Systems, Tenth Edition*. 10th edition. New York, N.Y.: McGraw-Hill Education. ISBN 978-1-25-964384-2.
- HONC, Daniel, 2018. *Automatizace 2*. Pardubice: Přednášky předmětu.
- JIANG, X.Y. a H. BUNKE, 1993. An optimal algorithm for extracting the regions of a plane graph. *Pattern Recognition Letters*. **14**(7), 553–558. ISSN 0167-8655. DOI: [https://doi.org/10.1016/0167-8655\(93\)90104-L](https://doi.org/10.1016/0167-8655(93)90104-L).
- MATT, J., 2021. spatialgraph2D. *MATLAB Central File Exchange* [online] [cit. 2021-04-12]. URL: <https://www.mathworks.com/matlabcentral/fileexchange/73630-spatialgraph2d>.
- RAK, Josef, 2017. *Teorie Grafů*. Pardubice: Univerzita Pardubice, FEI. Přednášky předmětu.
- RICHARD C. DORF, Robert H. Bishop, 2016. *Modern Control Systems*. 13th edition. Boston: Pearson. ISBN 978-0-13-440762-3.

- SCHWARZ, Douglas, 2021. Fast and Robust Curve Intersections. *MATLAB Central File Exchange* [online] [cit. 2021-04-12]. URL: <https://www.mathworks.com/matlabcentral/fileexchange/11837-fast-and-robust-curve-intersections>.
- SILVA, Guillermo J. – Aniruddha DATTA a S.P. BHATTACHARYYA, 2001. PI stabilization of first-order systems with time delay. *Automatica*. **37**(12), 2025–2031. ISSN 0005-1098. DOI: [https://doi.org/10.1016/S0005-1098\(01\)00165-0](https://doi.org/10.1016/S0005-1098(01)00165-0).
- TAN, Nusret, 2005. Computation of stabilizing PI and PID controllers for processes with time delay. *ISA Transactions*. **44**(2), 213–223. ISSN 0019-0578. DOI: [https://doi.org/10.1016/S0019-0578\(07\)90000-2](https://doi.org/10.1016/S0019-0578(07)90000-2).
- TAN, Nusret – Ibrahim KAYA – Celaledin YEROGLU a Derek P. ATHERTON, 2006. Computation of stabilizing PI and PID controllers using the stability boundary locus. *Energy Conversion and Management*. **47**(18), 3045–3058. ISSN 0196-8904. DOI: [doi:10.1016/j.enconman.2006.03.022](https://doi.org/10.1016/j.enconman.2006.03.022).
- VÍTEČKOVÁ M., Víteček A., 2011. *Vybrané metody seřizování regulátorů*. Ostrava: Vysoká škola báňská – Technická univerzita Ostrava. ISBN 978-80-248-2503-8.

SEZNAM PŘÍLOH

- A CD se zdrojovým kódem
- B Poznámky k implementaci v programu MATLAB

Příloha k diplomové práci

Návrh PI(D) regulátoru pro lineární systémy na základě automatického sestavení
oblasti stability

Bc. Marek Pakosta

CD se zdrojovým kódem

Příloha k diplomové práci

Návrh PI(D) regulátoru pro lineární systémy na základě automatického sestavení
oblasti stability

Bc. Marek Pakosta

Poznámky k implementaci v programu MATLAB

Oddíl 2.2 se zabýval detailním popisem jednotlivých kroků implementované metody HK. Tento oddíl se zabývá stručným popisem obecné struktury programu, který je součástí přílohy. SW realizace metody hraniční křivky byla implementována v jazyce MATLAB verze 2020a.

Metoda hraniční křivky je implementována pomocí zhruba 60 funkcí v cca 30 souborech a jednoho hlavního spouštěcího skriptu. Ukázky čtyř zdrojových kódů jsou uvedeny níže (viz zdroj. kód B.1 ÷ B.4). Všechny významné algoritmy jsou prostřednictvím pseudokódů popsány v průběhu praktické části diplomové práce.

Program se spouští pomocí skriptu `A_main`, ve kterém se definuje uvažovaný systém a potřebné parametry k výpočtu. Tento skript postupně volá funkci pro úpravu přenosu `normAndScale` (viz pododdíl 2.2.3), funkci pro výpočet průběhu HK `SBLcalc_ED` a funkci `F_getAllRegions`, která slouží k nalezení stabilních oblastí (viz pododdíl 2.2.6 ÷ 2.2.9) a v rámci nich také k nalezení optimálního nastavení (viz oddíl 2.3). Součástí skriptu `A_main` je i sada testovacích dynamických systémů, výběr je realizován odkomentováním příslušného řádku. Obecná struktura skriptu `A_main` (viz zdroj. kód B.1) je následující:

1. Inicializace parametrů a výběr přenosu systému – Mezi základní nastavitelné a volitelné parametry patří parametr:
 - `sat_lim` – velikost ohraničujícího obdélníku (mez saturace), viz obr. 2.13.
 - `f_rng` – rozmezí prokládání bodů vypočítané HK, prokládání je provedeno v případě, že `interpolate = 1`.
 - `grid_size` – určuje velikost matice, která představuje mřížku pro vyplnění oblasti body, viz obr. 2.51.
 - `kappa` – koeficient tlumení derivace regulované veličiny rozšířeného kritéria ISE (82).
 - `Td_min`, `Td_max` – minimální a maximální hodnota derivační časové konstanty T_D . Parametr `Td_num` určuje počet rovnoměrně rozložených hodnot mezi `Td_min` a `Td_max`.
 - `w_max_init` – počáteční ukončovací frekvence – nižší hodnota tohoto parametru zvýší počáteční přesnost ale způsobí také pomalejší výpočet a naopak.
2. Úprava přenosu systému (normalizace na kladné zesílení a případné rozšíření) pomocí funkce `normAndScale`. V případě, že je zvoleno `scale = 1`, tak je provedena

i úprava přenosu ve smyslu rozšíření dynamickým členem, viz pododdíl 2.2.3. To však v nejnovější verzi programu nemá význam.

3. Nalezení ukončovací frekvence pomocí funkce `F_getWc`, viz 2.2.2. Tu lze případně využít při použití jiné metody výpočtu. Možnost vykreslení AFFCh a FFCh pomocí funkce `F_Bde`.
4. Výpočet průběhu HK pomocí funkce `SBLcalc_ED`, viz pododdíl 2.2.1.
5. Volání funkce `F_getAllRegions`, která zajistí nalezení všech oblastí pro danou hodnotu T_D , z nichž lze následně snadno vybrat pouze ty stabilní – vstupem jsou některé výše uvedené parametry a výstupem je pole hodnot, které popisují průběh HK.
6. Grafická reprezentace oblastí stability spolu s odpovídajícím optimálním nastavením.

Funkce `F_getAllRegions` (viz zdroj. kód B.2) sestává z následujících částí:

1. Úprava vypočítaných hodnot popisujících průběh hraniční křivky (zvýšení rozlišení proložením bodů popisujících průběh HK, přidání uzavírací hrany (viz obr. 2.16)), pomocí funkce `F_HKnorm`.
2. Nalezení všech významných průsečíků pomocí funkce `F_getIntersections`, viz pododdíl 2.2.4.
3. Konstrukce grafů potřebných k reprezentaci průběhu HK a dále k reprezentaci jednotlivých oblastí pomocí funkce `F_createGraphs`, viz 2.2.5.
4. Reprezentace oblastí jako instance třídy `Oblast` (viz zdroj. kód B.3), viz pododdíl 2.2.6. Tomuto kroku předchází funkce pro rekonstrukci oblastí na základě konstruovaných grafů `F_getRegions` a funkce zajišťující získání významných bodů oblasti `F_vectorSum2` a `F_ziskejVB`.
5. Úprava reprezentace oblastí na základě jejich vzájemného topologického uspořádání pomocí funkce `F_Tpgy`, viz pododdíl 2.2.7.
6. Výplnění oblastí sítí bodů pomocí funkce `F_fill`, viz pododdíl 2.2.8.
7. Nalezení stabilních oblastí a v rámci nich i optimálních nastavitelných parametrů regulátoru (viz pododdíl 2.2.9) pomocí funkce `F_TS` (viz zdroj. kód B.4).

Zdrojový kód B.2 – Ukázka funkce F_getAllRegions

```

function [R1] = F_getAllRegions(x, y, Td, kappa, f_rng, interpolate, enclose,
    ↪ grid_size)
global tit N_pol D_pol h_Ggn h_Gvirt h_fill h_RP; warning('off'); %#ok<*ASGLU>
[x, y] = F_HKnorm(x, y, interpolate, enclose, f_rng); %úprava (interpolace, uzavření)
    ↪ průběhu HK
[out, nm] = F_getIntersections(x, y); %pomocí fce intersections.m
[G, tab, Gv, EC, out1, int_h] = F_createGraphs(out, x, y); [EC_rows, s] = size(EC); n
    ↪ = 0;
for i = 1:1:EC_rows %Pro všechny nalezené elementární smyčky
    [xr, yr, n_idx, h_lim] = F_getRegions(x, y, G, tab, nonzeros(EC(i, :))', out1);
    if h_lim == true
        continue %V případě, že je nějaká hrana hranice oblasti tvořena méně než 3
            ↪ body
    end
    NB = F_vectorSum2(xr, yr, n_idx);
    VB = F_ziskejVB(NB, xr, yr);
    R(n + 1) = Oblast(xr, yr, NaN, NaN, n_idx, NB, VB, 0, 0); n = length(R); close
        ↪ all;
end
R1 = F_Tpgy(R); [~, num_regions] = size(R1);
for i = 1:1:num_regions
    xr = R1(1, i).xt; yr = R1(1, i).yt; n_idx = R1(1, i).n_idx;
    [grx, gry] = F_fill(xr, yr, i, Td, grid_size);
    [S, opt] = F_TS(xr, yr, grx, gry, Td, kappa); %Určení stabilních oblastí a návrh
        ↪ PI(D) regulátoru
    NB = F_vectorSum2(xr, yr, n_idx);
    VB = F_ziskejVB(NB, xr, yr);
    R1(1, i) = Oblast(xr, yr, grx, gry, n_idx, NB, VB, S, opt);
    close all; %Řádek pro debugging umožňující zobrazení jednotlivých oblastí
end
end

```

Zdrojový kód B.3 – Ukázka třídy Oblast

```

classdef Oblast < handle
    properties
        xt;           %hranice oblasti v ose x
        yt;           %hranice oblasti v ose y
        grx;          %mřížka vnitřních bodů v ose x
        gry;          %mřížka vnitřních bodů v ose y
        n_idx;        %indexy vrcholů oblasti v poli xt, yt
        NB;           %nově vytvořené body
        VB;           %vnitřní body
        S;            %Booleovská proměnná rozhodující o stabilitě oblasti
        opt;          %pole parametrů optimalizace (pokud S=1)
    end
    methods
        function this = Oblast(xt, yt, grx, gry, n_idx, NB, VB, S, opt)
            this.xt = xt;
            this.yt = yt;
            this.grx = grx;
            this.gry = gry;
            this.n_idx = n_idx;
            this.NB = NB;
            this.VB = VB;
            this.S = S;
            this.opt = opt;
        end
        function znazorniOblast(this)
            par;
            plot(this.grx, this.gry, 'ro', 'MarkerSize', 3); hold on;
            plot(this.xt, this.yt, 'b-', LW, lw);
        end
        function znazorniNB(this)
            [r, ~] = size(this.NB); hold on;
            for i = 1:1:r
                plot(this.NB(i, 1), this.NB(i, 2), 'rx', 'MarkerSize', 15);
            end
        end
        function znazorniVB(this)
            [r, ~] = size(this.VB); hold on;
            for i = 1:1:r
                plot(this.VB(i, 1), this.VB(i, 2), 'rx', 'MarkerSize', 15);
            end
        end
    end
end
end
end

```

F_TS

```

function [S, opt] = F_TS(x, y, grx, gry, Td, kappa)
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Odeber vnitřní body v okolí HK:
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f_rng_local = abs(max(x)-min(x)); %adaptivní koeficient rozestupu
x = x(:); y = y(:); xy=[x';y']; f = cscvn(xy); [x, y, ~] = cubic_interp(f, f_rng_local);
↪ x = x(:)'; y = y(:)';

res = 25;
x_min = min(x); x_max = max(x); y_min = min(y); y_max = max(y);
tic; fx = abs(x_max-x_min)/res; fy = abs(y_max-y_min)/res;
for i = 1:1:length(x)
    for j = 1:1:length(grx)
        if grx(j)<x(i)+fx && grx(j)>x(i)-fx && gry(j)<y(i)+fy && gry(j)>y(i)-fy
            grx(j)=NaN; gry(j) = NaN;
        end
    end
end
grx = grx(~isnan(grx)); gry = gry(~isnan(gry));

global N_pol D_pol h_RP h_fill
minna = inf; kP_oa = inf; kI_oa = inf;
minnb = inf; kP_ob = inf; kI_ob = inf;
minnc = inf; kP_oc = inf; kI_oc = inf; t_end_Jc = 0;
minnd = inf; kP_od = inf; kI_od = inf; t_end_Jd = 0;
% stab = 0; nestab = 0; rpx_sn = zeros(1, length(grx)); rpy_sn = zeros(1,
↪ length(grx));
S = 1; t_end = 0; t1stb = tic;

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Stanov, zda-li je oblast nestabilní:
%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if isempty(grx)==1
    delta = -inf;
    S=0;
else
    idxs = randi([1, 10], 10, 1);
    for i = 1:1:length(idxs)
        kP = grx(idxs(i)); kI = gry(idxs(i));
        C1 = conv([Td*kP kP kI], N_pol); C2 = conv([1 0], D_pol); cs = coeff_sum(C1,
↪ C2);
        [~, delta] = getHurwitz(cs);
        if nnz(delta>0)~=length(delta) || any(real(roots(cs)) > 0)
            S = 0;
            break
        end
    end
end
end
end

```



```

%% %%%%%%%%%%%
%% V případě stabilní oblasti nalezní i optimální její nastavení:
%% %%%%%%%%%%%
if S == 1
    for i = 1:length(grx)
        kP = grx(i); kI = gry(i);
        C1 = conv([Td*kP kP kI], N_pol); C2 = conv([1 0], D_pol); cs = coeff_sum(C1,
            ↪ C2);
        Fwe = tf(C2, cs);
        Fwe.Denominator{1, 1} = conv(Fwe.Denominator{1, 1}, [1 0]);
        Fwy = tf(C1, cs);

        E1 = Fwe.Numerator{1, 1}; E2 = Fwe.Denominator{1, 1};
        % E1 = [6 0]; E2 = [1 6 11 12 1 0]; %Balate, str. 256
        [E1, E2] = upravPrenos(E1, E2);
        J1 = getJ(E1, E2);

        Y1 = conv(Fwy.Numerator{1, 1}, kappa); Y2 = Fwy.Denominator{1, 1};
        [Y1, Y2] = upravPrenos(Y1, Y2);
        J2 = getJ(Y1, Y2);

        Ja = J1;
        Jb = J1 + J2;
        if Ja <= 0 || Jb <= 0
            continue
        end
        if Ja < minna
            minna = Ja;
            kP_oa = kP; kI_oa = kI;
            Fwy_oa = Fwy;
        end
        if Jb < minnb
            i_opt = i;
            minnb = Jb;
            kP_ob = kP; kI_ob = kI;
            Fwy_ob = Fwy;
        end
    end
    opt = [kP_oa, kI_oa, kP_ob, kI_ob, minna, minnb, kappa, Td];
    drawOptSettings(opt, N_pol, D_pol, x, y, grx, gry);
    disp(sprintf('Testování stability dokončeno za %2.3f s', toc(t1stb)));
else
    opt = NaN*zeros(1, 8);
end
end
end

function [E1, E2] = upravPrenos(E1, E2)
NzLast_E1 = find(E1~=0, 1, 'last'); NzLast_E2 = find(E2~=0, 1, 'last');
E1_rem = length(E1)-NzLast_E1; E2_rem = length(E2)-NzLast_E2;
E1 = E1(1:length(E1)-min(E1_rem, E2_rem)); E2 = E2(1:length(E2)-min(E1_rem, E2_rem));
Nz1st_E1 = find(E1~=0, 1, 'first'); Nz1st_E2 = find(E2~=0, 1, 'first');
E1 = E1(Nz1st_E1:end); E1_len = length(E1); E2 = E2(Nz1st_E2:end); E2_len =
    ↪ length(E2);
E1 = [zeros(1, E2_len-E1_len-1), E1];
end

```

```

function J = getJ(E1, E2)
n = length(E2); Q = zeros(1, n-1);
for i = 1:1:n-1
    A = (-1)^(i-1);
    j = 1; sgn = 1; BC = E1(i)^2;
    while true
        sgn = sgn*(-1);
        if i-j > 0 && i+j < n
            BC_t = sgn*2*E1(i-j)*E1(i+j);
        else
            break
        end
        j = j + 1;
        BC = BC + BC_t;
    end
    Q(i) = A * BC;
end
[H, ~] = getHurwitz(E2);
H1 = [Q; H(2:end, :)];
J = (1/(2*E2(1)))*(det(H1)/det(H));
end

function [H, mnrs] = getHurwitz(pol)
p_n = length(pol)-1;
pol_even = pol(2:2:end); pol_odd = pol(1:2:end);
h1 = zeros(1, p_n); h1(1:length(pol_even))=pol_even;
h2 = zeros(1, p_n); h2(1:length(pol_odd))=pol_odd;

H = zeros(p_n, p_n); k = 1; mnrs = zeros(1, p_n-1);
for i = 1:1:p_n
    if mod(i, 2) == 1
        H(i, k:end) = h1(1:end-k+1);
    else
        H(i, k:end) = h2(1:end-k+1);
        k = k + 1;
    end
    if i < p_n
        mnrs(i) = det(H(1:i,1:i));
    end
end
end
end

```