



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV AUTOMATIZACE A INFORMATIKY

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

**ŘEŠENÍ SPOJITÝCH SYSTÉMŮ EVOLUČNÍMI
VÝPOČETNÍMI TECHNIKAMI**

DIZERTAČNÍ PRÁCE

DOCTORAL THESIS

AUTOR PRÁCE

AUTHOR

Ing. et Ing. Stanislav Lang

ŠKOLITEL

SUPERVISOR

doc. Ing. Radomil Matoušek, Ph.D.

BRNO 2017

ABSTRAKT

Práce je věnována problematice řešení spojitých systémů evolučními výpočetními technikami. Evoluční výpočetní techniky spadají do oblasti studia softcomputingu, jedná se o pokročilé optimalizační metaheuristiky, které se s postupným růstem výpočetního výkonu počítačů stávají stále více používanými metodami řešení složitých optimalizačních problémů. Řešení spojitých systémů, potažmo syntéza spojitých regulačních obvodů, je jednou z oblastí, kde tyto pokročilé algoritmy nalézají své uplatnění.

Při řešení spojitých systémů se zaměříme na problematiku regulace. Evoluční výpočetní techniky se pak mohou stát nástrojem nejen pro optimalizaci parametrů regulátoru, ale i pro návrh jeho struktury. Pro optimalizaci parametrů regulátoru lze využít mnoho různých algoritmů (genetický algoritmus, diferenciální evoluce, atd.), pro návrh struktury se setkáváme běžně s využitím tzv. gramatické evoluce. Nasazení gramatické evoluce však není nutné, pokud je využito vhodného kódování, jak je navrženo v předložené práci.

Práce prezentuje metodu návrhu struktury a parametrů obecného lineárního regulátoru s využitím genetického algoritmu. V rámci polynomiální teorie řízení se též setkáme s označením polynomiální regulátor. Způsob kódování popisu obecného lineárního regulátoru do genetického řetězce je stěžejní, určuje množinu algoritmů použitelných pro optimalizaci a ovlivňuje efektivitu výpočtů. Popsané kódování, efektivní implementace EVT, včetně multikriteriální optimalizace je stěžejním přínosem této práce.

KLÍČOVÁ SLOVA

řízení, regulace, regulátor, návrh, syntéza, optimalizace, kvalita regulace, obecný lineární regulátor, polynomiální regulátor, geometrické místo kořenu, optimalizační problém, kritériální funkce, paretooptimální řešení, Paretova křivka, Paretova hranice, metaheuristika, evoluční algoritmus, genetický algoritmus, GA, GAR, PID, 2DOF, ITAE

ABSTRACT

The thesis deals the issue of solution of continuous systems by evolutionary computational techniques. Evolutionary computing techniques fall into the field of softcomputing, an advanced metaheuristics optimization that is becoming more and more a method of solving complicated optimization problems with the gradual increase in computing performance of computers. The solution of continuous systems, or the synthesis of continuous control circuits, is one of the areas where these advanced algorithms find their application.

When dealing with continuous systems we will focus on regulatory issues. Evolutionary computing can then become a tool not only for optimization of controller parameters but also to design its structure. Various algorithms (genetic algorithm, differential evolution, etc.) can be used to optimize the parameters of the controller, for the design of the controller structure we usually encounter so called grammatical evolution. However, the use of grammatical evolution is not necessary if appropriate coding is used, as suggested in the presented thesis.

The thesis presents a method of designing the structure and parameters of a general linear controller using the genetic algorithm. A general linear regulator is known also as so called polynomial controller, if we encounter the polynomial theory of control. The method of encoding the description of the general linear controller into the genetic chain is crucial, it determines a set of algorithms that are usable for optimization and influence the efficiency of the calculations. Described coding, effective EVT implementation, including multi-criteria optimization, is a key benefit of this work.

KEY WORDS

control, controller, regulator, design, synthesis, optimization, quality of control, linear controller, polynomial controller, root locus, optimization problem, objective function, paretoptimal solution, Paret curve, Paret boundary, metaheuristics, evolutionary algorithm, genetic algorithm, GA, GAR, PID, 2DOF, ITAE

BIBLIOGRAFICKÁ CITACE

LANG, S. *Řešení spojitéch systémů evolučními výpočetními technikami*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 75 s. Vedoucí dizertační práce doc. Ing. Radomil Matoušek, Ph.D..

PROHLÁŠENÍ

Prohlašuji, že dizertační práci „Řešení spojitých systémů evolučními výpočetními technikami“ jsem vypracoval samostatně, za odborného vedení vedoucího dizertační práce. Dále prohlašuji, že jsem řádně citoval všechny použité prameny a literaturu.

V Brně, dne 13. 10. 2017

Ing. Stanislav Lang

PODĚKOVÁNÍ

Děkuji svým školitelům doc. Ing. Radomilu Matouškovi Ph.D. a doc. Ing. Ivanu Švarcovi CSc. za vedení při studiu, za spolupráci při výzkumu a za všechny čas mně věnovaný. Za cenné konzultace děkuji též doc. Ing. Zdeňku Němcovi CSc. V neposlední řadě chci poděkovat své manželce a rodině za podporu nejen při studiu.

V Brně, dne 13. 10. 2017

Ing. Stanislav Lang

Obsah:

	Abstrakt.....	3
	Abstract.....	5
	Bibliografická citace.....	7
	Úvod.....	15
1	Úvod do teorie automatického řízení.....	17
1.1	Systém a stabilita.....	17
1.2	Regulace.....	19
1.3	Prostředky regulace.....	20
1.4	Klasické a moderní regulátory.....	21
1.5	Kvalita regulace.....	24
2	Využití metaheuristik v optimalizaci.....	27
2.1	Vybrané heuristické metody a metaheuristiky.....	28
2.2	Evoluční algoritmy.....	31
2.3	Realizace genetického algoritmu (GA).....	33
2.4	Implementace GA v prostředí Matlab.....	35
3	Více-kriteriální optimalizace a Paretovo optimum.....	39
4	Testovací systavy.....	43
5	Využití EVT při návrhu regulátorů.....	45
5.1	Optimalizace parametrů PID s využitím GA.....	46
5.1.1	Koncepce a realizace.....	46
5.1.2	Optimalizace PID s využitím GA.....	47
5.2	Pareto-optimalizace parametrů PID s využitím GA.....	48
5.2.1	Motivace.....	48
5.2.2	Optimalizace regulátorů PID s jedním a dvěma stupni volnosti.....	48
5.2.3	Výsledky testování.....	50
6	Syntéza a optimalizace lineárních regulátorů.....	53
6.1	Klasické a moderní metody syntézy a optimalizace lineárních regulátorů.....	53
6.1.1	Obecný lineární regulátor a jeho optimalizace.....	53
6.1.2	Metoda geometrického místa kořene jako inspirace.....	54
6.1.3	Efektivita návrhu.....	55
6.1.4	Tvary operátorového přenosu.....	55
6.2	Vlastní metoda pro syntézu a optimalizaci lineárních regulátorů.....	56
6.2.1	Kriteriální funkce.....	56
6.2.2	Simulační model.....	56
6.2.3	Průběh optimalizačního procesu.....	56
6.2.4	Forma a princip kódování lineárních regulátorů.....	58
6.2.5	Úprava zesílení lineárního regulátoru.....	62
6.2.6	Výsledky syntézy GAR pro vybrané systavy.....	64
6.2.7	Výsledky syntézy GAR pro soustavu s dopravním zpožděním.....	67
7	Závěr.....	71
8	Seznam publikací autora.....	73
9	Seznam použité literatury.....	75

ÚVOD

Předložená práce spadá svým zaměřením do teorie automatického řízení, a rovněž do oblasti softcomputingu resp. moderních optimalizačních metaheuristik. Jak regulační technika, tak optimalizační metody zaznamenaly v posledních desítkách let značný rozvoj, čímž se otevřely nové možnosti, které klasická teorie řízení neznala.

Nutno podotknout, že zkušení regulační technici jsou poměrně konzervativní, jelikož průmysl vyžaduje ověřené a osvědčené metody zajišťující bezpečnost, robustnost a spolehlivost. Na druhé straně nalezneme obrovské nadšení mnohých, kteří objevili možnosti moderních výpočetních a optimalizačních metod, avšak v zápalu nadšení často odhodili mnohé poznatky teorie automatického řízení jako přežitě. Nedostatečné teoretické podložení moderních optimalizačních postupů činí tyto metody obtížně nasaditelnými do technické praxe, jelikož jsou často opomenuta omezení reálných soustav (totéž ale platí i pro některé klasické postupy). Jindy jsou moderní postupy použitelné, ale málo efektivní.

Diskutován bude veliký potenciál, který nabízí prostor mezi uvedenými disciplínami, respektive jejich propojení. Práce tak představí jeden z možných přístupů efektivního využití moderních postupů optimalizace v automatizaci.

Autor navazuje disertací na svoji diplomovou práci *Optimalizace řídicího algoritmu pomocí evolučního algoritmu* [3], ve které se věnoval optimalizaci parametrů regulátorů PID v jejich spojitě a diskrétní variantě. Od optimalizace parametrů pevně dané struktury PID regulátoru se posuneme k optimalizaci PID s dvěma stupni volnosti a především k návrhu obecného lineárního regulátoru.

Postupů pro návrh obecného lineárního regulátoru, též označovaného jako polynomiální regulátor, existuje celá řada. Některé metody vycházejí z kořenového hodografu, jiné provádí návrh ve frekvenční oblasti, nalezneme i analytické metody (např. metoda optimálního modulu). Vzhledem k často kladeným multikriteriálním požadavkům se stále více rozšiřují metody založené na simulaci. Zde nalézají široké využití optimalizační metody. Setkáme se s klasickými gradientními metodami, optimalizačními heuristikami, i pokročilými metaheuristikami často inspirovanými přírodními zákony, či chováním živočišných společenstev.

Práce se při návrhu a optimalizaci regulačních obvodů vydává cestou nasazení evolučních výpočetních technik. Je však zároveň demonstrováno, jak zásadní vliv má volba vhodného kódování optimalizačního problému pro efektivitu optimalizace. Vhodné kódování totiž umožní syntézu obecného lineárního regulátoru, včetně návrhu struktury, s využitím klasických genetických algoritmů, aniž by bylo zapotřebí využívat gramatickou evoluci. Druhým milníkem práce je diskuse výsledků vícekriteriální optimalizace, jelikož v praxi je zvolený způsob řízení často kompromisem mezi požadavky více kritérií.

1 ÚVOD DO TEORIE AUTOMATICKÉHO ŘÍZENÍ

1.1 Systém a stabilita

„Systém je uspořádaná množina prvků a vztahů mezi nimi, které z určitého souboru tvoří relativní celek.“ [33] Samotný pojem systém je natolik abstraktní, že proniká napříč mnoha obory od teoretické matematiky, přes biologii, ekonomii až po technologické procesy.

Systémy můžeme rozdělit na systémy bez dynamiky a systémy s dynamikou. Systémy bez dynamiky dále dělíme na systémy bez paměti (např. saturace) a systémy s pamětí (např. hystereze) [5]. Větší zájem však budeme soustředit na systémy s dynamikou. I zde můžeme zavést dělení na dvě základní kategorie, a to tzv. t -variantní systémy (parametry systému se mění v čase) a systémy t -invariantní, jejichž parametry jsou časově nezávislé.

Popis systému lze provádět různými způsoby, z nichž nejobecnější a prakticky vždy použitelný je popis pomocí diferenciálních rovnic v případě systému spojitého, respektive rovnic diferenčních v případě diskrétního systému. Popis systému lze kategorizovat na *vnitřní* a *vnější* popis. Převod vnitřního popisu na vnější popis je jednoznačný. Převod vnějšího popisu na vnitřní popis není jednoznačný, protože záleží na volbě stavových veličin. [4, 5, 6, 7, 8, 11, 12]

Vnitřní popis umožňuje jednoznačně popsat vnitřní strukturu systému, vede na tzv. *stavový model systému*, který je tvořen množinou vstupů, množinou vnitřních stavů, množinou výstupů, množinou funkcí popisujících vazby vstupů a stavů na stavy a množinou funkcí popisujících vazby vstupů a stavů na výstupy. Pro obecný (nelineární) t -variantní systém platí vztah [5]:

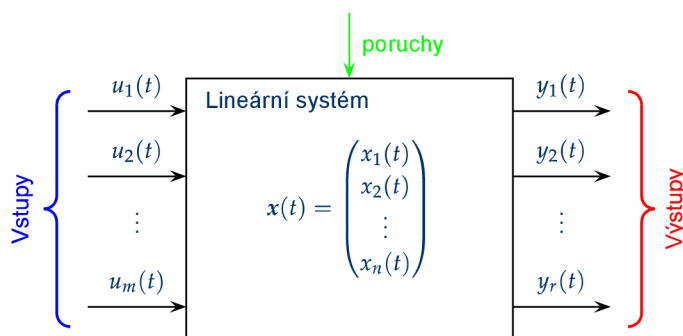
$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \\ \mathbf{y}(t) &= \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t)\end{aligned}\quad (1)$$

kde je \mathbf{x} vektor stavových proměnných, \mathbf{u} vektor hodnot vstupů, \mathbf{y} vektor hodnot výstupů, t parametr času, \mathbf{f} vektor nelineárních závislostí určujících derivace stavových proměnných jako funkce stavových veličin a vstupů, \mathbf{g} vektor nelineárních funkcí popisujících výstupní signály.

Pro lineární t -variantní systémy lze užít jednodušší vztah:

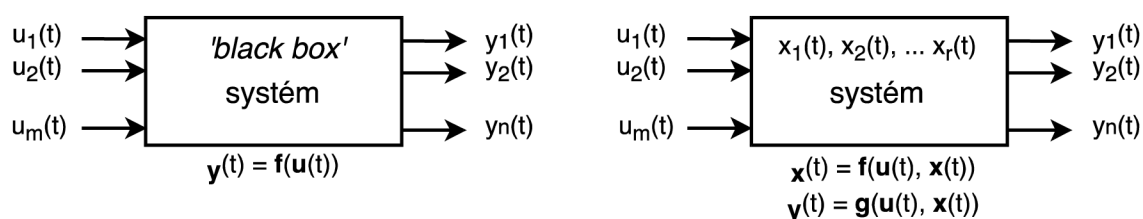
$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C} \mathbf{x}(t) + \mathbf{D} \mathbf{u}(t)\end{aligned}\quad (2)$$

kde je \mathbf{x} vektor stavů, \mathbf{u} vektor vstupů, \mathbf{y} vektor výstupů, \mathbf{A} matice dynamiky, \mathbf{B} matice vazeb vstup – stav, \mathbf{C} matice vazeb stav – výstup a \mathbf{D} matice přímých vazeb vstup – výstup



Obr. 1 Obecný systém [4]

Alternativou k vnitřnímu popisu je popis vnější, který však popisuje systém pouze pomocí vztahů mezi jeho vstupy a výstupy. Co se odehrává uvnitř systému vnější popis nezachycuje, neboť na systém pohlíží jako na tzv. *black box* a vnitřní strukturu ignoruje. Výhodou je vyšší stupeň abstrakce, jelikož vnitřní struktura již není relevantní a všechny systémy se stejným vnějším chováním lze popsat identicky. V důsledku ztráty informací a absence znalosti vnitřních stavů není možné řešení systémů s nenulovými počátečními podmínkami (příkladem z technické oblasti mohou být nenulové stavy vnitřních akumulátorů energie).



Obr. 2 Vnější a vnitřní popis systému

Významnou vlastností, na jejímž základě lze systémy rozdělovat, je jejich linearita. [20] Lineární dynamické systémy jsou popsány lineárními diferenciálními rovnicemi, případně soustavou lineárních diferenciálních rovnic nižšího stupně. Základní vlastností lineárních systémů je platnost principu *superpozice*, tj. platí že odezva systému na signál složený z více složek je totožná jako součet odezev na jednotlivé složky budícího signálu. Lineární systémy mají pouze jeden ustálený stav. Naproti tomu nelineární systémy platnost pravidla superpozice nezachovávají a rovněž připouštějí více ustálených stavů, a to nejen v podobě singulárního bodu (jak je tomu u systémů lineárních), ale též v podobě tzv. *periodických řešení*. Blíže se problematikou nelineárních systémů zabývá [5, 11].

Řešení lineárních systémů je výrazně jednodušší a existuje celá řada popisů alternativních k popisu diferenciálními rovnicemi, což ve většině případů zjednodušuje práci řešitele. V praxi se setkáme s popisem pomocí přenosů v Laplaceově transformaci či přenosů frekvenčních, dále jsou využívány grafické popisy pomocí přechodových, impulzních nebo frekvenčních charakteristik, v neposlední řadě jmenujme popis systému pomocí rozložení pólů a nul. Mezi všemi uvedenými formami popisu lze volně přecházet, aniž by došlo ke ztrátě informace o chování systému. Pokud bychom podmínky linearity hodnotili velmi přísně, shledali bychom, že v reálném světě se setkáváme převážně s nelineárními systémy, neboť téměř všechny systémy vykazují saturaci, a tudíž princip superpozice neplatí v neomezeném rozsahu. Na druhou stranu nic nebrání tomu, abychom k systémům přistupovali jako k lineárním, jestliže se chovají lineárně v pracovní oblasti. Dokonce i u systémů které vykazují v pracovní oblasti nelineární chování provádíme často tzv. linearizaci v pracovním bodě, abychom s nimi mohli pracovat v určitém rozsahu pracovní oblasti jako s lineárními, aniž bychom se dopustili větších odchylek od skutečného (nelineárního) chování.

Další podstatnou vlastností systému případně jeho popisu je skutečnost, zda je systém spojité či diskrétní. Mohli bychom rozlišit *spojitost v čase* a *spojitost v hodnotě*, i když ve většině případů jdou tyto skutečnosti ruku v ruce. Mluvíme-li o spojitém systému, máme většinou na mysli systém pracující ve spojitém čase, a analogicky diskrétní systém, jako systém pracující v diskrétních časových okamžicích.

Matematický popis diskrétních systémů [4, 21] se od popisu spojitého systému mírně liší, nicméně popisy i postupy řešení jsou většinou analogické. Místo diferenciálních rovnic využíváme rovnice diferenční, místo Laplaceovy transformace pak využíváme Z-transformaci. Novým pojmem pro diskrétní systémy je tzv. perioda vzorkování, což jsou okamžiky, ve kterých jsou známy hodnoty vstupů a výstupů, eventuálně stavů.

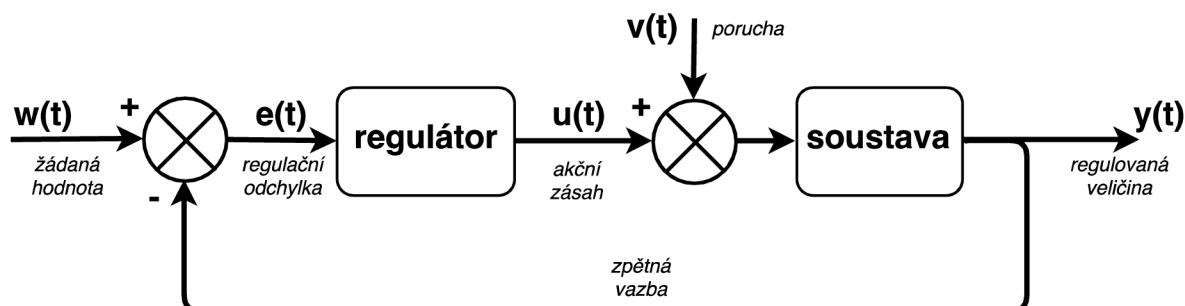
Stabilita je pojem, který je se systémy úzce svázán. Neopustíme proto teorii systémů, aniž bychom se o stabilitě zmínili, neboť se jedná prakticky o základní vlastnost systému. Jedním z nejobecnějších přístupů chápání pojmu *stabilita systému* je stabilita ve smyslu *BIBO* (ohraničený vstup, ohraničený výstup) [20], což znamená, že na vstupní signál, který je omezený v amplitudě a čase, soustava odpoví výstupním signálem, který je rovněž omezený v amplitudě a čase. Jestliže budeme stabilitu ve smyslu *BIBO* aplikovat na vnitřní popis systému, je toto kritérium přísnější, omezený musí být v hodnotě a čase nejen každý z výstupů, ale též každý z vnitřních stavů, tj. hodnota žádného stavu nesmí růst nad všechny meze [7, 21]. Budeme-li vyšetřovat stabilitu systému ve stavovém prostoru, nabízí se Ljapunovova definice stability, zde rozlišujeme lokální stabilitu rovnovážného stavu, lokální asymptotickou stabilitu a globální stabilitu (stabilitu ve velkém) [5].

1.2 Regulace

Pojem *regulace* není v moderním světě ani zdaleka neznámým pojmem. Méně známa je již skutečnost, jak velké množství regulačních pochodů se v našem okolí odehrává. Pokud bychom zůstali u intuitivního pojetí regulace, mohli bychom prohlásit, že každá rostlinná či živočišná buňka má zavedeno mnoho regulačních pochodů pro udržení optimálních podmínek ke svému fungování, rozmnožování atd. Obdobné příklady bychom našli nejen v mikrosvětě, ale i v klasickém světě u různých ekosystémů, či v makrosvětě. *Otázky po původu inteligence vložené do uvedených systémů spadají do oblasti teologie a filozofie a zde se jim nebudeme věnovat.*

V teorii automatického řízení se pojem regulace používá ve smyslu zpětnovazebního řízení dynamických systémů. V oblasti řízení dynamických systémů dále rozlišujeme tři úrovně řízení: ovládání (řízení bez zpětné vazby), regulaci (řízení se zpětnou vazbou) a vyšší formy řízení, tj. řízení s inteligentními, optimálními či adaptivními znaky. Ovládání představuje v průmyslu asi pouze 6% aplikací řízení (jelikož se jedná o řízení bez zpětné vazby, mění se vlastnosti přenosu žádané hodnoty na výstup, ale neeliminují se vlivy poruch). 85% veškerého řízení tvoří regulace [28, 29]. Zbýlých 9% pak tvoří vyšší formy řízení [28, 29].

Základním prvkem regulace je zpětná vazba, tj. informace o skutečné hodnotě regulované veličiny. Mluvíme-li o zpětné vazbě, máme na mysli prakticky výhradně zápornou zpětnou vazbu. Rozdíl žádané hodnoty a regulované veličiny určuje regulační odchylku. Regulační odchylka, případně její vývoj v čase, jsou informací, na jejímž základě generuje regulátor akční zásahy. Existují však i případy kladné zpětné vazby, která je většinou silně nežádoucí (například v elektroakustice), najdeme však i příklady úmyslného zavedení kladné zpětné vazby (například u komparátorů). „Norbert Wiener, považovaný za zakladatele kybernetiky, přirovnával zpětnovazební smyčku ke slepecké holi, která dává slepci zpětnou informaci o jeho pohybu a ovlivňuje tak jeho pohyb následující.“ [34].

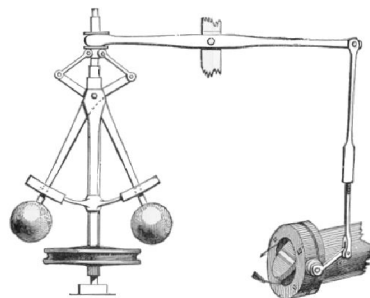


Obr. 3 Regulační obvod

1.3 Prostředky regulace

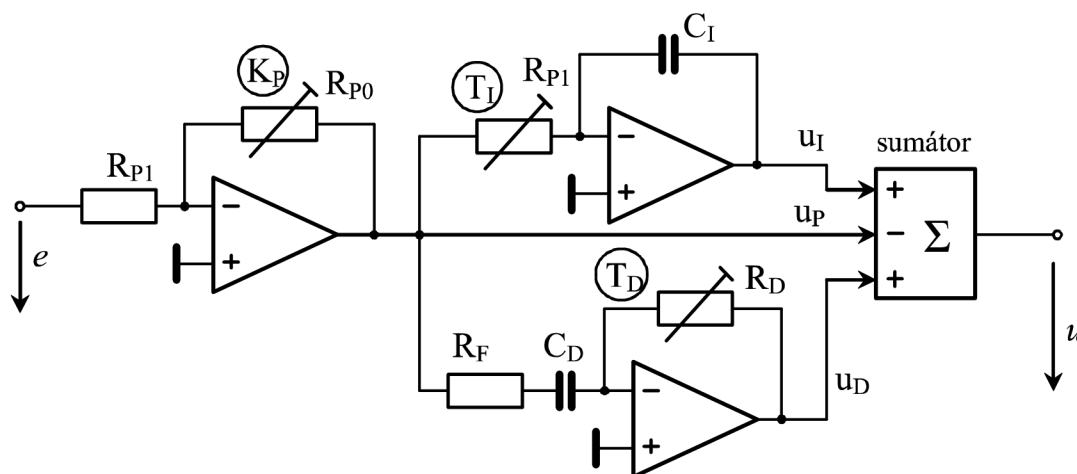
První zmínky o regulaci spadají již do období starověku. „Už od starodávna používali mlynáři na vodních a větrných mlýnech jednoduché zařízení, které regulovalo přísun zrní mezi mlýnské kameny v závislosti na jejich otáčkách.“ [15]

Literatura často označuje za první regulátor (novodobého pojetí) *Wattův odstředivý regulátor* (viz ilustrace), který měl za úkol zajistit konstantní otáčky parního motoru při proměnlivém zatížení. Zaveden byl roku 1782.



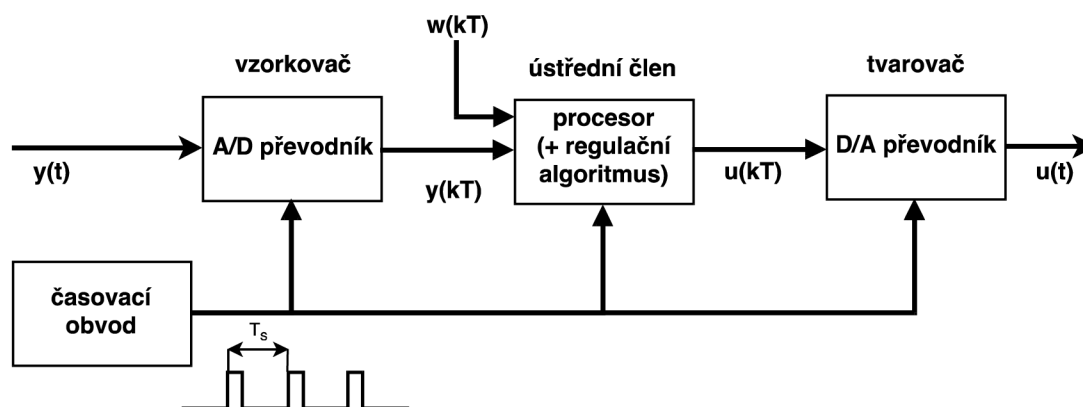
Obr. 4 *Wattův odstředivý regulátor* [15]

Největší rozvoj regulační techniky probíhá ve dvacátém století. K první polovině 20. století datujeme zavedení regulátoru PID („*derivační složka použita poprvé v roce 1935*“ [34]). Regulátory ve tvaru PID mají dodnes majoritní podíl na regulaci v průmyslu, různé literární zdroje uvádí procentuální zastoupení mezi 85% a 95%. Až do nedávné doby byly regulátory založeny na operačních zesilovačích a pracovaly ve spojitém čase.



Obr. 5 *Realizace PID regulátoru s operačními zesilovači* [10]

Vývoj řídicí techniky však ve dvacátém století neskončil, naopak nabyl na rychlosti. S prudkým vývojem výpočetní techniky v posledních desetiletích se posunuly i možnosti regulační techniky výrazně dopředu. Výpočetní jádro regulátorů je tvořeno mikroprocesorem. Na rozdíl od spojitě pracujících operačních zesilovačů pracují mikroprocesory s časem diskretním a regulátor tak musí být doplněn o obvody zajišťující diskretizaci dat na vstupu (vzorkovač, tj. A/D převodník) a obvody převádějící digitální hodnoty zpět na spojitou veličinu (tvarovač, tj. D/A převodník).



Obr. 6 Číslicový regulátor

Pro číslicové regulátory je zásadní perioda vzorkování, respektive její poměr ke konstantám regulovaného spojitého systému. Jestliže je perioda vzorkování řádově podobná časovým konstantám, pohlížíme na regulátor jako na diskretní. Je-li však perioda vzorkování o několik řádů nižší než časové konstanty systému, můžeme se velmi přiblížit vlastnostem regulátoru spojitého. Způsob nastavování parametrů pro výše uvedené varianty regulátorů je pak odlišný.

1.4 Klasické a moderní regulátory

Zaměříme-li se na samotné regulátory, jejich provedení a principy činnosti, naskytne se mnoho způsobů dělení a kategorizování. Můžeme je rozčlenit na základě fyzikálního principu, na němž jsou založeny. Odlišili bychom tak regulátory mechanické, hydraulické, pneumatické a elektronické, přičemž elektrické regulátory mohou být hybridní a mluvíme pak například o regulátorech elektro-pneumatických či elektro-hydraulických.

Z pohledu teorie řízení je ale zajímavější princip fungování regulátoru a jeho vnitřní struktura, než konečná fyzická realizace. Stejně jako fyzická realizace regulátoru musí být „ušita“ na míru fyzikální podstatě a požadavkům na fungování regulovaného systému, tak i formu regulačních pochodů je třeba vhodně vybrat. Nelze proto obecně stanovit, který z regulátorů je více či méně vhodný, zda PID či fuzzy nebo dvupolohová regulace, takové prohlášení můžeme učinit až na základě zhodnocení vlastností regulované soustavy. Uvedme tedy nejvýznamnější skupiny.

- **Dvupolohová regulace**

Jedná se o nejjednodušší způsob regulace založené na principu zapnuto/vypnuto. Dvupolohová regulace je nelineární a má nejrozmanitější formy fyzické realizace.

Ačkoli jde zdánlivě o poměrně primitivní způsob regulace, je pro některé aplikace naprosto dostačující [10]. Pro některé regulované systémy nemá smysl uvažovat sofistikovanější regulátory, neboť by efektivitu regulace nezvýšily. Příkladem může být regulace teploty v žehličce, kde regulaci zajistí bimetalový pásek, nebo regulace hladiny ve splachovadle, kde je zpětná vazba zajištěna přes plovák.

Dvupolohovou regulaci je vhodné volit v případech, kdy je regulovaný systém charakterizován velkými časovými konstantami (v porovnání s rychlostí reakce regulátoru). Často je regulátor doplněn o nelinearitu typu *pásmo necitlivosti* či *hystereze*, aby bylo zabráněno příliš častému spínání a rozpínání v okolí rovnovážného stavu a je tak chráněn akční člen proti rychlému opotřebení.

Avšak ne každý regulátor, který připouští na svém výstupu pouze dvě hodnoty (zapnuto/vypnuto) spadá do kategorie dvupolohových regulátorů. U elektronických regulátorů z jiných tříd se totiž můžeme setkat s tzv. pulzně šířkovou modulací (PWM) na výstupu regulátoru. PWM předpokládá vysokou frekvenci signálu, tak aby perioda byla výrazně menší (o několik řádů)

než časové konstanty systému. V takovém případě bude mít soustava vůči signálu dolnoproústvní filtrační charakter a akční zásah bude odpovídat průměrné hodnotě signálu.

- **Regulátory PID**

Patří k nejstarším regulátorům a v průmyslové praxi jsou nejrozšířenějšími regulátory vůbec. Jejich nasazení a rozvoj lze datovat již k první polovině 20. století. Pomineme-li P regulátory, objevily se jako první hydraulické regulátory ve tvaru PI [35], derivační složka byla vyvinuta v roce 1935 [35], v roce 1939 pak představila firma Taylor svůj pneumatický regulátor, který již strukturálně odpovídal tvaru PID [36]. Následně elektronické regulátory PID byly založeny na operačních zesilovačích, v dnešní době jsou již realizovány pomocí mikroprocesorové techniky. Kromě modifikací P, I, PI, PD a PID se objevily další strukturální modifikace ve formě PI-D či I-PD nebo ještě obecnější 2DOF PID. Další doplňky či strukturální modifikace pak řeší filtraci derivační složky a potlačení windup efektu. V průmyslu mají dle některých zdrojů regulátory PID 85% podíl, podle jiných dokonce až 95% podíl na řízení procesů. K. J. Åström uvádí, že 97% regulátorů v chemickém průmyslu vykazuje tvar PID [26].

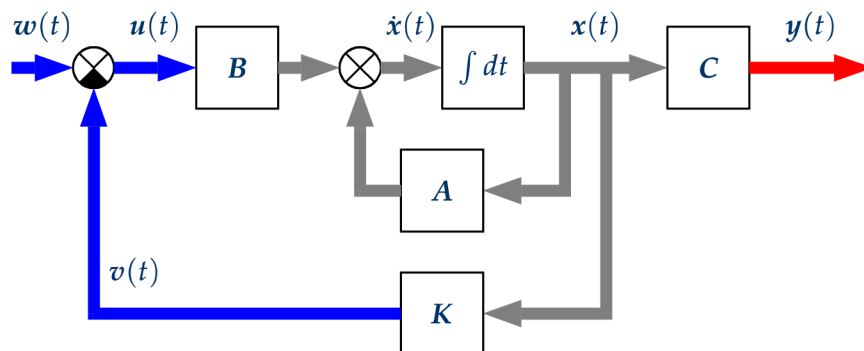
Máme-li stručně popsat základní vliv jednotlivých složek PID na regulační děj, lze říci, že velká proporcionální složka P zajišťuje rychlý regulační děj, ale při působení poruchy způsobuje velkou chybu. Integrační složka I se stará o zajištění nulové ustálené odchylky řízení, a to i při působení poruchy, snižuje však celkovou stabilitu soustavy a zpomaluje přechodový děj. Pomocí derivační složky D jsme naopak schopni dosáhnout větší stability a urychlení přechodového děje, negativním vlivem mohou být příliš rychlé změny akčního zásahu a zesilování šumu, proto je u derivační složky realizováno *filtrování derivační složky* [7], které tyto negativní vlivy snižuje.

- **Obecné lineární regulátory**

Regulátor PID lze popsat jako lineární systém pomocí lineárních diferenciálních rovnic. Rovněž strukturální modifikace PID budou lineární, pokud se v nich nebudou vyskytovat nelinearity, jako například saturace u integrátoru pro potlačení windup efektu nebo na výstupu pro omezení akčního zásahu. Množina lineárních regulátorů však nabízí větší možnost změny dynamiky regulační soustavy než její podmnožina PID. Pro návrh regulátorů je možno využít celou řadu výpočetních či grafických metod. [4, 11] Lineárním regulátorům se na tomto místě nebudeme více věnovat, neboť lineární regulátory jsou hlavním předmětem zájmu této práce a věnovat se jim budeme v následujících kapitolách.

- **Stavové regulátory**

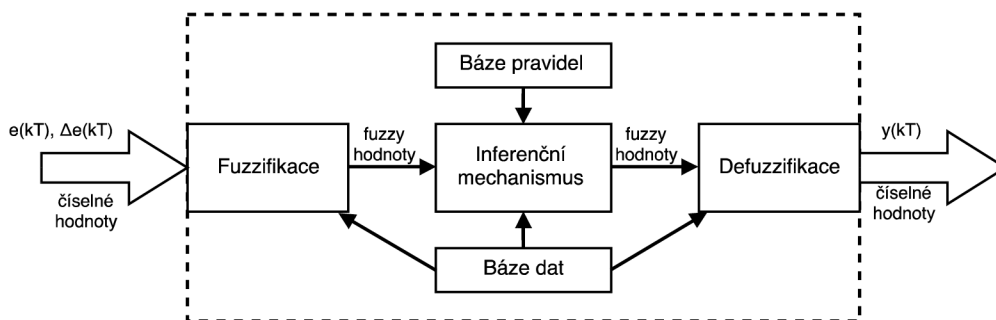
Stavový regulátor zavádí zpětné vazby od všech či pouze vybraných stavů, rozlišujeme tak úplnou a částečnou stavovou regulaci. Aby mohla stavová regulace fungovat, je zapotřebí aby byly stavy měřitelné, nebo alespoň rekonstruovatelné. Stavovou regulaci uvažujeme pro lineární systémy, zpětné vazby mají proporcionální charakter. Pomocí vhodného nastavení regulátoru lze dosáhnout teoreticky libovolné změny dynamiky. Pro výpočet regulátoru se nejčastěji využívá tzv. metod *pole placement* [4, 8], které umožňují modifikovat geometrická místa kořenů (pólů). Změna dynamiky je však v praxi omezena, protože větší dynamika předpokládá větší akční zásahy do soustavy. Velikost akčních zásahů je pak v praxi omezena co do přípustnosti a realizovatelnosti.



Obr. 7 Stavová regulace [8]

- **Fuzzy regulátory**

Fuzzy regulátory jsou založeny na zcela odlišném mechanismu než regulátory popsané výše. Nejsou již popsány diferenciálními rovnicemi, ale pomocí pravidel, která jsou aplikována inferenčním mechanismem s různou vahou. Místo klasických ostrých číselných hodnot je využíváno tzv. fuzzy lingvistických hodnot (málo, moc apod.) s určením míry příslušnosti. O převod číselné hodnoty regulační odchylky na fuzzy hodnoty se stará proces fuzzifikace na vstupu regulátoru. Analogicky je vypočtený akční zásah (interně reprezentovaný lingvistickými hodnotami) převeden na výstupu regulátoru pomocí tzv. defuzzifikace na číselnou hodnotu. Fuzzy regulátory jsou obecně nelineární a jejich nasazení je vhodné zejména pro systémy s neurčitostmi a pro nelineární systémy. [11, 37]



Obr. 8 Fuzzy regulátor

- **Neuronové sítě**

Regulátory založené na neuronových sítích tvoří zcela speciální třídu regulátorů. Na rozdíl od ostatních skupin regulátorů nejsou parametry nastavovány zvenčí, nýbrž neuronová síť si je nastavuje sama během procesu *učení*. Rozlišujeme dvě základní skupiny neuronových sítí, a to síť s čistě dopředným šířením informace (forward propagation), a síť se zpětným šířením informace (back propagation), přičemž prve jmenovaná varianta nedisponuje pamětí, kdežto druhá ano. Informace v neuronových sítích nejsou soustředěny do konkrétních celků struktury, ale jsou rozptýleny po celé síti. Absence možnosti převodu naučené sítě na exaktní matematický popis činí volbu neuronových regulátorů pro běžné systémy nevýhodnou. Na druhou stranu, neuronové sítě jsou schopny řídit i procesy, které nelze matematicky dobře popsat, a jsou též vhodné pro nelineární systémy, které vykazují nestabilitu požadovaného ustáleného stavu. Jinými slovy, neuronové sítě je vhodné použít tam, kde ostatní typy regulátorů selhávají, nebo nedokáží zajistit dostatečnou kvalitu regulace. Například v [38] lze nalézt srovnání neuronového regulátoru s PID pro úlohu řízení aktivního magnetického ložiska.

- **Obecné nelineární regulátory**

Moderní mikroprocesorová technika umožňuje vytvářet regulátory s dynamikou nejrůznějších podob, a to i s nelineárními závislostmi. Regulátory mohou být navrhovány na základě matematického popisu regulované soustavy s využitím *linearizace vstup-výstup* případně *linearizace vstup-stav* [5]. Nebo může být pro vygenerování obecného regulátoru využito metaheuristik. Efektivně lze využít metod gramatické evoluce nebo transplantační evoluce [32, 35], která je kombinací evoluce gramatické a diferenciální.

- **Optimální regulátory**

Za zmínku stojí *optimální regulátory*, kdy pojem může být poněkud zavádějící, neboť neuvažujeme extrémální řešení vůči volně stanovenému kritériu, nýbrž vůči kritériu minimalizace času transportu tělesa z bodu *A* do bodu *B*. „Potřeba řešení této úlohy vznikla v 50-tých letech minulého století, kdy při letech do vesmíru bylo potřebné především vyřešit optimalizační problém s minimalizací paliva spotřebovaného v raketě.“ [7]

- **Adaptivní regulátory**

Na závěr výčtu uvedme ještě skupinu adaptivních regulátorů. Nejedná se již o třídu, která by zapadala do výše uvedeného dělení. Principiálně se může jednat o regulátory nejrůznějších tvarů a provedení, od PID, lineárních a fuzzy až po nelineární regulátory. Základní vlastností uvedených

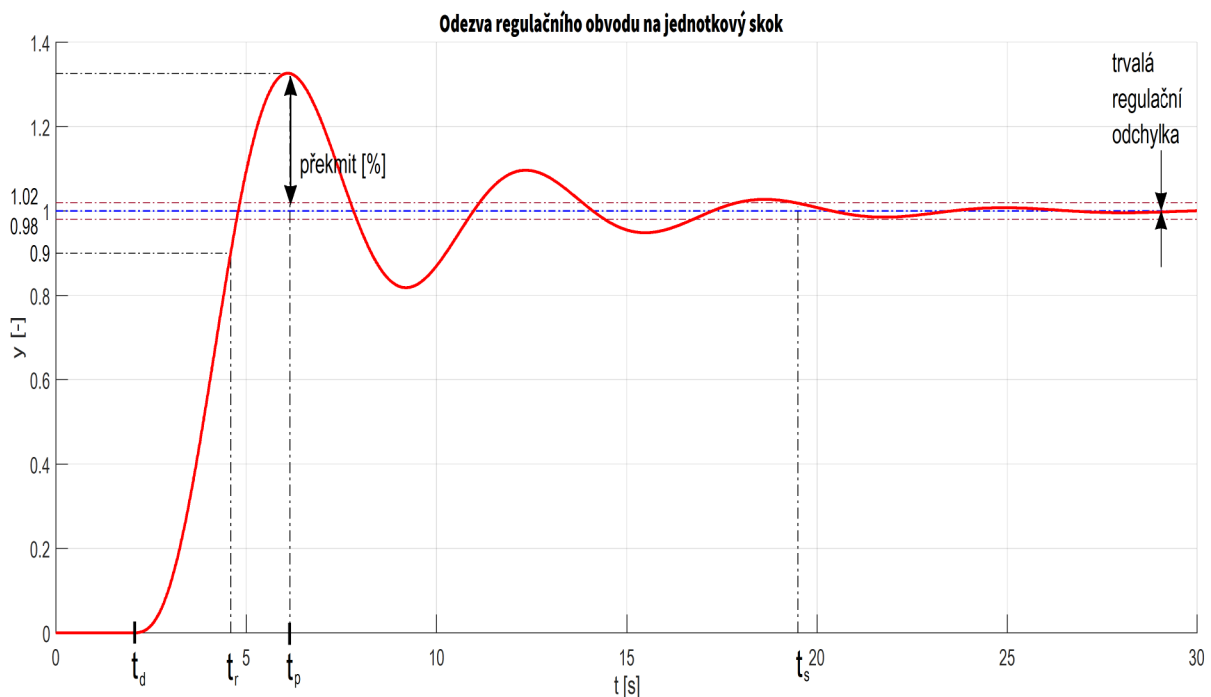
regulátorů je možnost adaptace, tj. schopností měnit své parametry případně strukturu na základě změn parametrů řízeného systému. Změnami v regulačním obvodu mohou být např. změna pracovního bodu, změna parametrů soustavy či změna vlivu prostředí (poruchy, atd.). Adaptace může být skoková či plynulá, realizovaná za pomoci heuristik, případně se může jednat o adaptivní regulátory s modelem systému či soustavy.

1.5 Kvalita regulace

Dříve, než se budeme věnovat jednotlivým kritériím ovlivňujícím kvalitu regulace, je na místě zmínit již uvedený pojem *stabilita*. Stabilitu můžeme nahlížet v pojetí BIBO (bounded input bounded output) [20], což znamená že odpověď systému na signál omezený v hodnotě a čase bude opět omezená (v hodnotě a čase). Nicméně více nás možná bude zajímat stabilita ustáleného stavu systému. Definice stability ustáleného stavu je více, mluvíme o stabilitě lokální a globální, stabilitě asymptotické atd. Pro lineární systémy jednotlivé definice stability splývají, neboť je-li systém stabilní, je globálně asymptoticky stabilní, stručně řečeno, vždy konverguje k jedinému globálnímu ustálenému stavu. U systémů nelineárních je situace komplikovanější, neboť regulační obvod (výsledný systém) může disponovat větším množstvím ustálených stavů a cyklických řešení, z nichž některá mohou být stabilní a jiná nestabilní a jednotlivá řešení mohou mít různá okolí přitažlivosti. [4, 11, 13]

Stabilitu by bylo z jistého úhlu pohledu možno brát za jednu z kvalit regulace, více ji však chápeme jako předpoklad k tomu, abychom o kvalitě regulace mohli vůbec začít mluvit.

Ukazatele kvality jsou determinovány požadavky na regulační děj. Kvalitu pak nejčastěji posuzujeme v časové nebo frekvenční oblasti. Určování kvalit regulace v časové oblasti se nejčastěji odvíjí od odezvy systému na skok žadané hodnoty (přechodová charakteristika systému), případně odezvy systému na skok poruchy. Parametry odezvy, které nás budou zajímat, mohou být: maximální překmit, doba přeběhu (většinou 10 – 90%), mrtvá doba, doba náběhu, doba průtahu, doba ustálení v tolerančním pásmu, doba prvního, případně dalších maxim, u systémů s neminimální fází podkmit. Frekvenční oblast pak bude zajímavá zejména pro lineární systémy, kde můžeme pro obvod určit amplitudovou bezpečnost, fázovou bezpečnost, případně bezpečnost v modulu. Potažmo můžeme diskutovat vlastnosti obvodu v jednotlivých frekvenčních pásmech.



Obr. 9 Základní sledované parametry přechodového děje

Dále budeme při posuzování regulačních dějů vycházet z kvalit regulace v časové oblasti. Jak bylo již lehce nastíněno, parametrů určujících kvalitu může být skutečně mnoho. Jestliže má dojít k optimalizaci na základě více kritérií, stojíme náhle před problémem multikritériální optimalizace [9, 22]. Postup při řešení vícekritériálního problému můžeme být dvojí: můžeme stanovit váhy jednotlivých kritérií a vybudovat tak jedno složené kritérium, nebo můžeme hledat tzv. *pareto-optimální řešení* (viz. následující kapitoly).

překmit <i>overshoot</i>	Překmit nastává, jestliže výstupní veličina překročí žádanou hodnotu. Vyskytuje se zejména při přechodovém ději, často doprovázen kmitáním.
t_d dopravní zpoždění <i>delay</i>	Dopravní zpoždění způsobuje zpoždění reakce systému a celkově komplikuje proces regulace (zejména pokud je jeho velikost srovnatelná s velikostí dominantních časových konstant soustavy).
t_r doba přeběhu <i>rise time</i>	Doba přeběhu je čas potřebný pro přechod výstupní veličiny z $x\%$ na $y\%$ žádané hodnoty. <i>V této práci je dolní hranice nastavena na 0% a horní hranice na 90%.</i>
t_p doba překmitu <i>peak time</i>	Vrchol <i>peak</i> je nejvyšší hodnota dosažená při přechodovém ději před dosažením žádané hodnoty, <i>peak-time</i> je čas dosažení vrcholu.
t_s doba ustálení <i>setting time</i>	Doba ustálení výstupní veličiny je čas, který uplyne mezi okamžikem skokové změny žádané hodnoty na vstupu soustavy a okamžikem kdy se výstupní veličina ustálí na žádané hodnotě v pásmu tolerance . <i>V této práci používáme toleranční pásmo $\pm 2\%$</i>

Tabulka 1: Vybrané parametry přechodového děje

Pro konkrétní zadání úlohy však budeme uvažovat jenom jedno optimum. Jestliže výslednou kvalitu regulace určuje více ukazatelů, bude se velmi často jednat o hledání kompromisů, jelikož mnohé požadavky na regulaci mohou být až protichůdné. Jedním z příkladů, kdy jsou požadavky do značné míry protichůdné je regulace kmitavých soustav, kdy chceme minimalizovat dobu ustálení a minimalizovat překmit. Existují případy, kdy je například naprosto nepřijatelný překmit (pohyb robotického ramene, apod.) jelikož by mohlo dojít k poškození či zničení produktu či stroje samotného. V takovém případě se z těchto požadavků stávají *de facto omezující podmínky* pro optimalizaci dalších parametrů.

Jelikož je vícekritériální optimalizace složitá, byla vyvinuta kritéria, která v sobě s různou vahou sdružují jednotlivé požadavky. Za nejznámější a nejpoužívanější lze brát integrální kritéria kvality regulace [11, 13]. Integrální kritéria mají za úkol minimalizovat integrál časového průběhu regulační odchylky, přičemž průběh odchylky je jednotlivými kritérii před integrací vhodně upraven. Jako první z integrálních kritérií uveďme *kvadratické integrální kritérium (ISE)*.

$$J_K = \int_0^t e^2(t) dt \quad (3)$$

Uvedené kritérium nemusí dosahovat nejlepších výsledků, je však výhodou, že optimalizace je proveditelná i analyticky, viz [11], což je však vhodné jen pro menší počet optimalizovaných parametrů, jelikož složitost vztahů pak rychle narůstá.

Dalšími kritérii jsou (usměrněné) lineární integrální kritérium (IAE), integrální kritérium

ITAE (integrál odchylky násobené časem), případně kritérium ITSE, které pro účely integrace využívá kvadrátu regulační odchylky násobeného časem. Všechna uvedená kritéria jsou uvedena v tabulce níže.

Za nejpoužívanější lze považovat integrální kritérium ITAE, které klade větší důraz na rychlost ustálení přechodového děje. Výpočet je prováděn numericky za pomoci výpočetní techniky.

Zkratka	Název kritéria	Vzorec
ISE	Kvadratické kritérium (<i>Integral of Squared Error</i>)	$f_{ISE} = \int_0^t e^2(t) dt$
IAE	Lineární (usměrněné) kritérium (<i>Integral of Absolute Error</i>)	$F_{IAE} = \int_0^t e(t) dt$
ITSE	Kritérium ITSE (<i>Integral of Time multiply Squared Error</i>)	$f_{ITSE} = \int_0^t t e^2(t) dt$
ITAE	Kritérium ITAE (<i>Integral of Time multiply Absolute Error</i>)	$f_{ITAE} = \int_0^t t e(t) dt$

Tabulka 2: Běžně používaná integrální kritéria

Dosud nezmiňným kritériem posuzování kvality může být monotónnost přechodových dějů. Pro analytický návrh parametrů regulátoru se pro tyto účely používá například metody optimálního modulu, která vychází z frekvenčních charakteristik (přičemž se omezuje na lineární systémy).

Robustnost regulace je velmi často skloňovaným termínem. Robustnost patří též ke kvalitám regulace a v poslední době je na ni kladen poměrně velký důraz při diskuzi kvality [25]. Stejně jako mnoho dalších pojmů je i robustnost chápána poměrně intuitivně. Obecně ji lze popsat jako schopnost obvodu (celkového systému) zachovat stabilitu a pokud možno co nejlepší kvalitu regulace i při změně okolností za kterých systém pracuje. Uvažujeme jednak robustnost ve smyslu odolnosti vůči působení poruch. Druhým pohledem na robustnost pak může být odolnost vůči změnám parametrů regulované soustavy. V případě působení neurčitostí v parametrech soustavy lze pro lineární systémy využít Kharitonovy polynomy, kdy je posouzena stabilita všech krajních řešení (která lze reprezentovat jako vrcholy n -rozměrného kvádrů vymezeného mezemi jednotlivých parametrů), a jestliže jsou všechny kombinace stabilní, můžeme konstatovat stabilitu pro zkoumaný systém s působením neurčitostí.

Dosud jsme se soustředili pouze na posuzování průběhu regulované veličiny. Důležitými však jsou často i jiné průběhy v regulačním obvodu, a to zejména průběh akčního zásahu. Velmi často je kladen důraz na plynulost a nekmitavost akčních zásahů, aby tak nedocházelo k zbytečnému opotřebení akčního členu. Nutno též brát v potaz, že akční člen je omezen co do velikosti a často i rychlosti akčního zásahu. Proto je nutno vždy zvážit, zda jsou pro danou optimalizaci průběhy akční veličiny realizovatelné a vhodné. Většinou je to právě akční člen, který do obvodu vnaší nelinearitu saturace, těmto vlivům se budeme rovněž věnovat v následujících kapitolách.

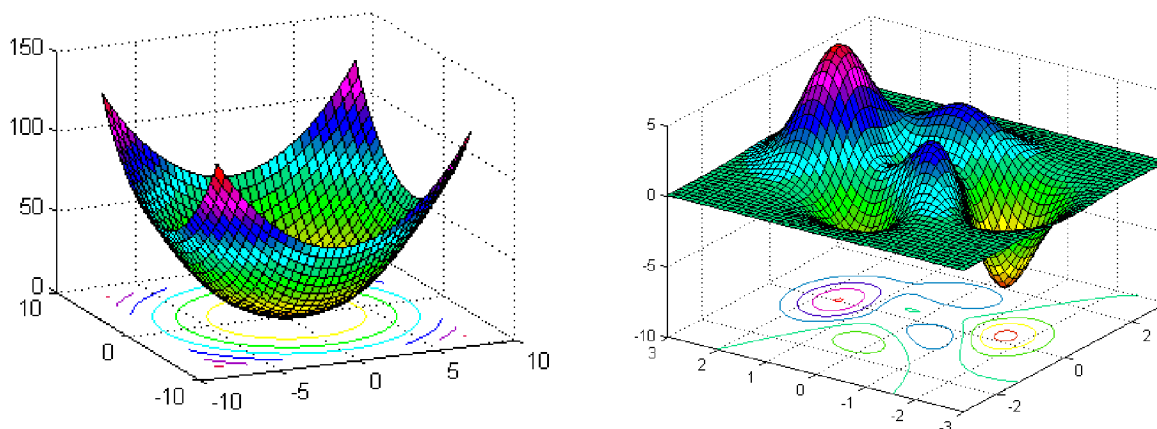
2 VYUŽITÍ METAHEURISTIK V OPTIMALIZACI

Jestliže jsme mluvili v předchozí kapitole o regulaci zprvu velmi obecně a nalézáme první stopy zmíněné disciplíny již ve starověku, mohli bychom jít v rámci pojmu *optimalizace* ještě dále. Nejen člověk, ale i každý živý organismus upravuje své vnitřní pochody i vnější projevy tak, aby mohl přežít a dobře fungovat v daných podmínkách prostředí. Že se u jednotlivých typů organismů, dokonce i jedinců mohou některé projevy za různých podmínek lišit, je závislé na různé stavbě a nastavení konkrétního organismu, tj. různém umístění optima, případně rozdílné podobě hodnotící funkce (kriteriální funkce). Kriteriální nebo též účelová funkce je matematický popis problému, který má být optimalizován. Optimalizace úlohy pak spočívá zpravidla v hledání globálního extrému. Řešení kriteriální funkce lze ve zjednodušeném tvaru popsat následujícím vztahem:

$$\begin{aligned} \mathbf{x}_{opt} &= \arg \text{opt } f(\mathbf{x}) \\ \mathbf{x} &\in X^m \end{aligned} \quad (4)$$

Dostáváme se k řešení matematických úloh infinitesimálního a variačního počtu ve funkcionálních prostorech [39], nebo k nasazení numerických metod. Přestože se disciplína matematiky vyvíjí již několik tisíciletí a učinila nesmírné pokroky vedoucí nejen k efektivnímu řešení úloh, ale i k rozvoji lidského myšlení a logiky, zůstává řešení vybraných typů úloh stále neuspokojivé a některé úlohy nejsou standardními postupy řešitelné vůbec. Díky prudkému rozvoji výpočetní techniky začaly koncem minulého století vznikat pokročilé heuristické metody optimalizace, které hledaly inspiraci ve funkci a chování jedinců a společenstev mikroorganismů a vyšších organismů, případně v přírodních pochodech. Aby byla zdůrazněna hlubší myšlenka a sofistikovanost heuristik, byl připojen prefix *meta*, mluvíme tedy o metaheuristikách. Nelze pochybovat o skutečnosti, že tyto přírodou inspirované metody optimalizace byly ověřeny tisíci, možná i milióny let, neboť pouze organismy schopné se neustále přizpůsobovat podmínkám a znovu a znovu hledat optimální podmínky pro svou existenci mohly přežít. Nasazení těchto metod pak ukázalo, že pro vybrané úlohy je dosahováno velmi uspokojivých řešení, a to i tam, kde řešení dříve nebylo ani představitelné. Nutno podotknout, že stejně jako různé organismy či společenstva aplikují odlišné metody pro hledání optima, tak i nasazení optimalizačních heuristik silně závisí na typu řešené úlohy, a je tedy silně problémově závislé.

Jak bylo již řečeno, jednotlivé typy optimalizačních úloh se mohou výrazně lišit v obtížnosti řešení. Jestliže bude funkce disponovat jediným extrémem a bude spojitá, nalezení řešení bude možné pomocí klasických metod, a nasazení metaheuristik absolutně nemá smysl a je kontraproduktivní. Jestliže však je úloha více-dimenzionální a obsahuje lokální extrémy, případně je v některých oblastech nespojitá nebo i nedefinovaná, pak klasické metody často selhávají a pole je připraveno pro nasazení vhodné metaheuristiky.

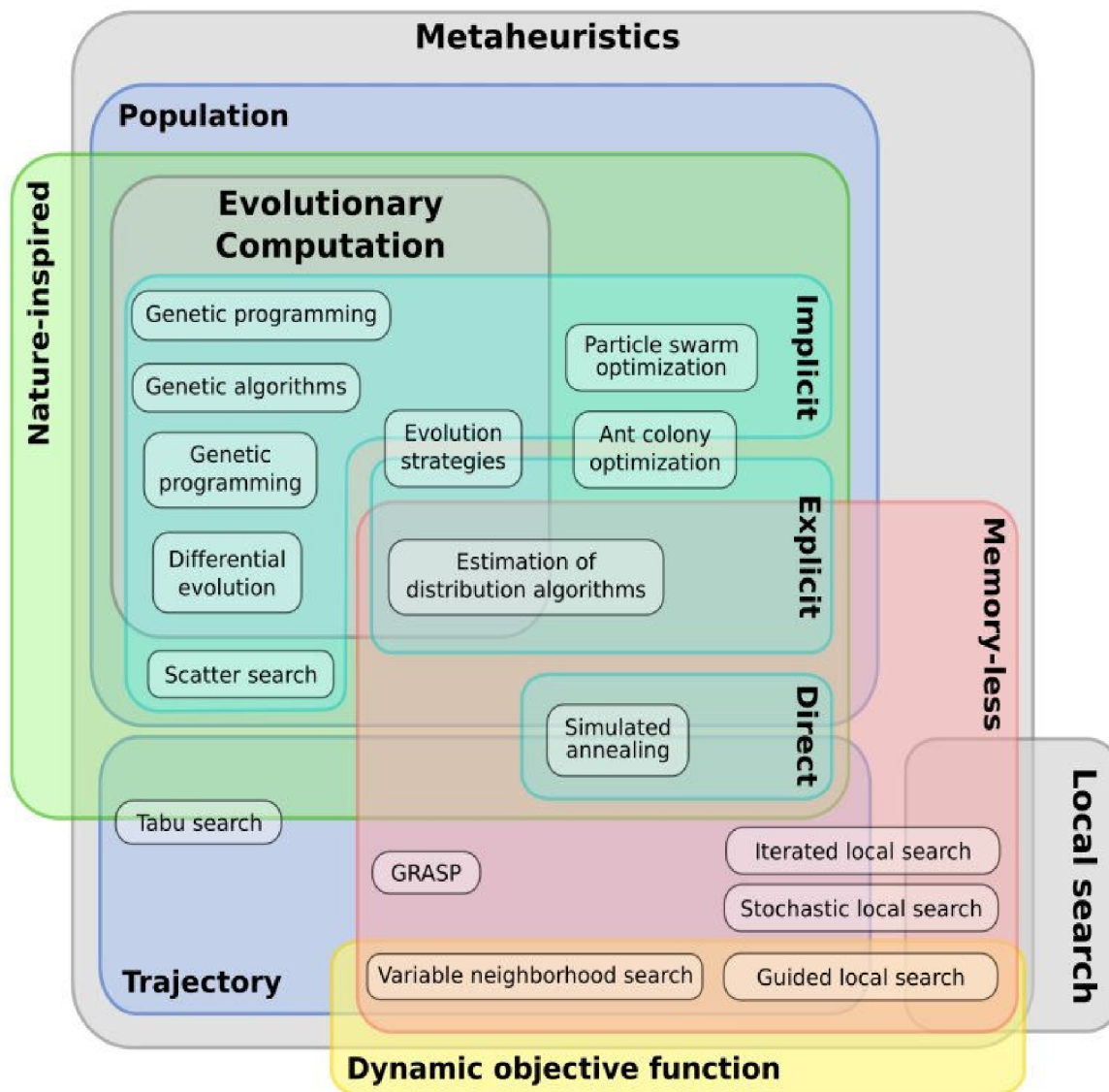


Obr. 10 Ilustrační obrázek funkcí dvou proměnných [3]

2.1 Vybrané heuristické metody a metaheuristiky

Množina metaheuristik je velmi početná, neboť zmíněných optimalizačních algoritmů je celá řada a přibývají stále nové, přičemž většina z nich je hybridní implementací již existujících metaheuristik, objevují se však i zcela nové algoritmy. Existence velkého množství (více či méně se navzájem lišících) metaheuristik je důsledkem skutečnosti, že tyto metody jsou silně problémově orientované. Nalezneme tak poměrně jednoduché, ale i vysoce sofistikované optimalizační metody. Tam, kde jedna metoda pokulhává, nebo je málo efektivní, jiná může dosahovat výborných výsledků, a naopak.

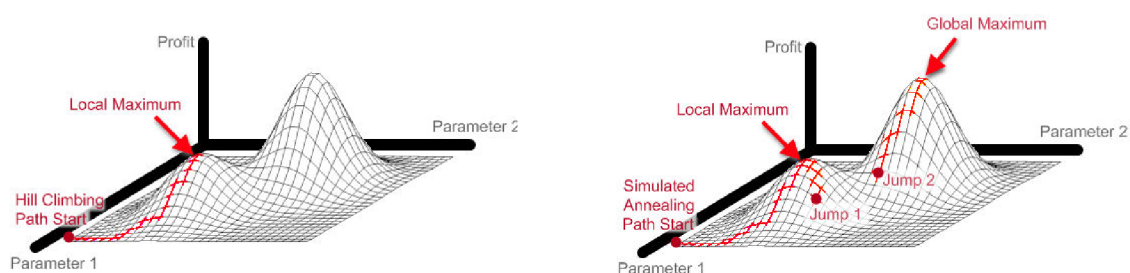
Existuje celá řada vlastností, na základě nichž je možné jednotlivé metaheuristiky členit do skupin. Zásadním kritériem je skutečnost, zda metoda pracuje s jedním řešením v rámci iterace, nebo s tzv. populací (tj. množinou řešení, kdy tato množina zahrnuje více než jedno řešení v rámci iterace). Nalezneme však řadu dalších faktorů pro dělení. Následující obrázek uvádí jednu z nejznámějších kategorizací:



Obr. 11 Možné rozdělení metaheuristik [40]

V následujících odstavcích krátce představíme vybrané metaheuristiky, které patří k nejvýznamnějším, nejedná se však o plný výčet. Cílem je, aby si nezasvěcený čtenář učinil představu o rozmanitosti jednotlivých metod. Více informací k uvedeným algoritmům lze nalézt například v publikaci *Evoluční výpočetní techniky: Principy a aplikace* [9].

Jako první uveďme **metodu lokálního hledání** (LocalSearch), která je bezesporu heuristickou metodou optimalizace, zda ji zařadit do skupiny metaheuristik je ale sporné. Průběh optimalizace v rámci jedné iterace je zcela deterministický, neboť zlepšující řešení jsou hledána výhradně ve směru nejvyššího gradientu, nicméně metoda opakuje hledání v rámci iterací z náhodně volených počátečních řešení. Problémem metody je vysoké riziko uvíznutí v lokálním extrému, pokud funkce není monotónní.



Obr. 12 Ilustrace srovnání lokálního a globálního řešiče [18]

Výrazným vylepšením metody lokálního hledání je **horolezecký algoritmus** (Hill Climbing), který prohledává okolí současně dosaženého nejlepšího řešení. Metoda tak připouští v rámci jednoho kroku i zhoršující řešení. Nicméně je tato metoda silně závislá na volbě velikosti okolí a přetrvává zde vysoké riziko uváznutí v lokálním extrému (tentokrát ve formě oscilace mezi dvěma body). Algoritmus proto bývá různými způsoby modifikován (horolezecký algoritmus s učením, paralelní horolezecký algoritmus, atd., viz [9]).

V osmdesátých letech minulého století na horolezecký algoritmus navázal F. Glover s **metodou zakázaného prohledávání** (Tabu Search), kdy byl horolezecký algoritmus doplněn o krátkodobou (v některých případech i dlouhodobou) paměť. Je tak výrazně sníženo riziko zacyklení, nicméně velikost paměti je novým parametrem, který je nutno vhodně zvolit, jelikož krátká paměť (zakázaných řešení) snižuje efekt potlačení rizika zacyklení, naopak příliš dlouhá paměť může způsobit „přeskočení“ oblasti s globálním extrémem.

Z výše uvedené rodiny algoritmů uveďme ještě **metodu simulovaného žihání** (Simulated Annealing). V roce 1983 ji představil S. Kirkpatrick [41], který se nechal inspirovat fyzikálními ději probíhajícími v metalurgii při žihání kovů. Při žihání tuhého tělesa dochází ke stabilizaci krystalické mřížky kovu, a dosahuje se tak jeho lepších vlastností. Při vysoké teplotě se připouštějí stavy, kdy částice zaujme i energeticky méně výhodnou pozici, s klesající teplotou je pravděpodobnost přechodu k horšímu řešení stále méně pravděpodobná. Podobně i algoritmus zprvu připouští i zhoršující řešení a s klesajícím argumentem *teploty* (teplota je řídicí parametr algoritmu) se snižuje pravděpodobnost výskytu zhoršujících řešení a metoda se chová podobně jako horolezecký algoritmus. Připouštění zhoršujících řešení pomáhá metodě vymanit se z lokálních extrémů a zvýšit tak pravděpodobnost nalezení globálního extrémálního řešení.

Zcela odlišnou kategorii optimalizačních metaheuristik tvoří hejnové algoritmy. Jedná se o mnoho modifikací vycházejících z jedné myšlenky. Jedním z nejznámějších algoritmů je **metoda rojení částic** (Particle swarm optimization, PSO), vyvinutá Dr. Eberhartem a Dr. Kennedym v roce 1995. Algoritmy pracují již s tzv. populacemi, tedy celými soubory řešení v rámci jedné iterace. Jednotlivá řešení jsou inicializována náhodně a jednotliví *jedinci* (konkrétní řešení) se pohybují v n-dimenzionálním prostoru směrem k současně nejlepšímu řešení. Inspiraci metody lze hledat v přírodě v chování různých živočišných společenstev, např. ptačích hejn (metoda ptačího hejna byla prezentována již o čtyři roky dříve). Modifikací metody rojení částic a hejnových algoritmů je celá řada, více [9, 42].

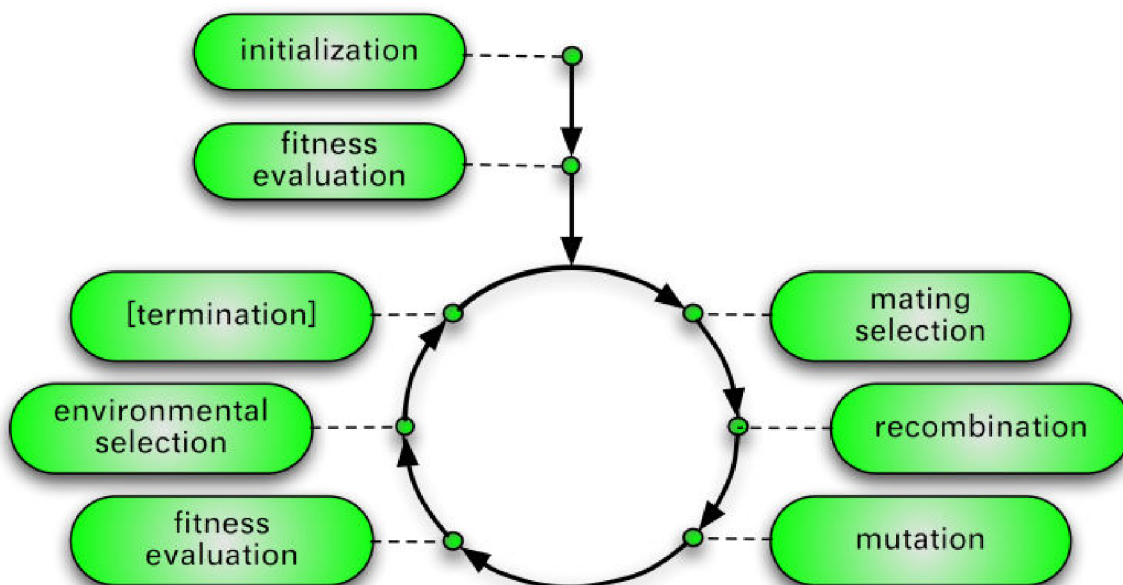
Inspiraci ve fungování živočišných společenstev našla i metoda **optimalizace mravenčí kolonií** (Ant colony optimization, ACO), prezentovaná v roce 1992 J. Dorigem, viz [9]. Myšlenka je následující: mravenci hledající potravu se pohybují náhodně prostorem. Jestliže některý jedinec nalezne zdroj potravy, vrací se do mraveniště zanechávaje za sebou feromonovou stopu. Feromon začne ovlivňovat pohyb ostatních mravenců a mravenci jsou přitahováni na cestu k potravě. Čím více mravenců se po cestě k potravě pohybuje, tím výraznější je stopa a dosah feromonu je větší. Jakmile je však zdroj potravy vyčerpán, stane se cesta neatraktivní, jelikož feromon postupně vyprchá. Skutečnost, že účinek feromonu s časem slábne, je důležitá nejen proto, aby staré a již neatraktivní stezky nemátly jedince v jejich pohybu, ale je faktorem umožňujícím efektivní optimalizaci. Kratší, a tedy efektivnější cesty jsou proběhnuty za kratší časový interval, stopa tak zůstane déle než stopa vedoucí delší cestou. Stejně tak, pokud se na trase objeví překážka, začnou ji jednotliví jedinci obcházet z různých stran, nejkratší cesta je projita za nejkratší dobu a stopa na ní je tedy výraznější, a postupem času se tak všichni jedinci vydají optimální trasou. Uvedená metaheuristika nalézá své uplatnění především v problémech vedoucích na hledání cesty grafem a její optimalizaci. Využití nalezneme například v optimalizaci routování v počítačových sítích, apod.

Poněkud zvláštní skupinou jsou tzv. **samo-organizující se migrační algoritmy** (SOMA), které stojí na pomezí evolučních algoritmů, ale stále spadají do oblasti algoritmů hejnových (swarm optimization). Jedná se opět o algoritmy inspirované v přírodě zejména živočišnými společenstvy, podobně jako ACO. Ať už se inspirujeme rozličnými druhy (predátory, včelami, atd.), většinou se bude jednat o hledání potravy, přičemž jedinci vybaveni inteligencí spolu navzájem komunikují a kooperují. Zaujímají tak v prostoru různě se měnící uskupení, rozdělují se a spojují, odtud samoorganizace. Algoritmy SOMA používají vektorové operace nad jedinci (řešeními) migrujícími koordinovaně prostorem, jeden optimalizační cyklus nazýváme *migrační kolo* [9].

Přicházíme k poslední skupině našeho výčtu, a to skupině vysoce sofistikovaných metaheuristik, kterými jsou **Evoluční algoritmy** (EA). Těmto optimalizačním metodám je věnována následující podkapitola. Podotkneme znovu, že výše zmíněný výčet metaheuristik zahrnuje pouze nejvýznamnější metody a nelze jej zdaleka brát za vyčerpávající, ale spíše orientační. Výstižný popis nejen uvedených metod včetně nastínění programové realizace lze dohledat v publikaci *Evoluční výpočetní techniky: Principy a aplikace* [9].

2.2 Evoluční algoritmy

Evoluční algoritmy tvoří početnou skupinu optimalizačních metaheuristik, nejznámější jsou *genetický algoritmus* (GA), *evoluční strategie* (ES), *genetické programování* (GP), *gramatická evoluce* (GE), *diferenciální evoluce* (DE), ale nalezneme řadu dalších hybridních a kombinovaných řešení, např. Transplantační evoluce [31, 32, 35].



Obr. 13 Cyklus evolučního algoritmu [44]

Pro všechny evoluční algoritmy je typické použití operací **křížení**, **selektce** a **mutace**. Uvedené pojmy jsou analogií k biologickým procesům uplatňovaným v genetice. Ostatně celá myšlenka evolučních algoritmů vychází z inspirace genetikou. Není pochyb, že genetika je geniální, tisíce a miliony let ověřenou metodou optimalizace, díky níž jsou organismy adaptabilní vůči změnám prostředí. Počátky studia genetiky nacházíme v 19. století. Za zakladatele oboru genetiky je považován přírodovědec Gregor Johann Mendel (1822-1884), augustiniánský mnich, který se věnoval zkoumání dědičnosti (zejména na rostlinách). Dalším významným jménem je Charles Darwin (1809-1882), který je zakladatelem evoluční biologie. *Evoluční teorii opíral o přírodní výběr a pohlavní výběr* [45], a v současné době je i jednou z obecně uznávaných teorií o vzniku živočišných druhů. Principy přírodního a pohlavního výběru lze pozorovat nejlépe u nižších organismů (např. bakterií), odtud jsou inspirovány operátory selektce jedinců v rámci algoritmů genetické optimalizace.

Dříve než přistoupíme k stručnému představení jednotlivých algoritmů, uveďme základní pojmy užívané v teorii evolučních algoritmů:

- **Jedinec** – Pojem *jedinec* označujeme jedno konkrétní řešení optimalizačního problému. Jako genotyp jedince označujeme (zpravidla binární) reprezentaci kódovanou v genetickém řetězci (pozn.: způsob kódování má zásadní vliv na proveditelnost a efektivitu optimalizace), jako fenotyp následně označíme jedince ve smyslu reprezentace pro konkrétní řešený problém.
- **Populace** – Množina jedinců, se kterými pracujeme v rámci jednoho optimalizačního cyklu, tj. jedné generace.
- **Generace** – nahrazuje pojem iterace, jedná se o jeden optimalizační cyklus.
- **Fitness** – *Fitness* nebo též *vhodnost* je určením míry kvality jedince. Jako fitness tedy budeme označovat hodnotu účelové funkce pro konkrétního jedince.
- **Výběr** – Je-li použito slovo *výběr* samostatně, máme zpravidla na mysli *výběr pro účely křížení*. Vybíráme tak jedince, kteří budou vzájemně kombinováni. Metod výběru je celá řada,

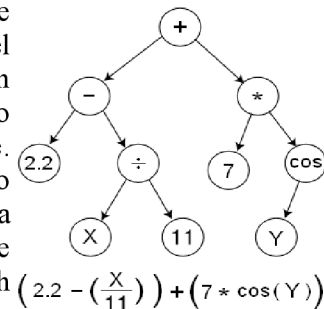
např. stochastický, poměrový, turnajový, ruletový.

- Výběr prostředí – Jedná se o výběr jedinců pro účely dalšího kola optimalizace, tj výběr jedinců do následující generace. Může být aplikován tzv. elitismus, kdy je vždy uchováno nejlepší řešení, nebo jsou vybráni pouze nově vygenerovaní jedinci, atd.
- Křížení – je operací, kdy dochází ke kombinaci parametrů původních jedinců a vzniku jedinců nových. V podstatě napodobujeme chování, které lze sledovat u bakterií, které si vymění část své genetické informace. Křížení zpravidla zachovává délku genetického řetězce, ale není to podmínkou. Křížení může být jednobodové či vícebodové.
- Mutace – je náhodná změna jedné či více hodnot v genetickém řetězci jedince. V závislosti na metodě je mutace uplatněna před operací křížení či po operaci křížení. Způsobů nastavení pravděpodobnosti mutace je opět celá řada. Příliš vysoká míra mutace by navodila víceméně chaotický proces optimalizace a degradovala by metodu prakticky na náhodné hledání (random walking), naopak pokud by míra mutace byla příliš nízká, mohlo by dojít k malé různorodosti jedinců a stagnaci vývoje.

Genetický algoritmus (GA) vešel v širokou známost v roce 1975, kdy jej ve své publikaci *Adaptation in Natural and Artificial Systems* představil Johan Holland [9, 43, 46]. Historie genetického algoritmu však sahá mnohem hlouběji, a to do padesátých let 19. století, kdy Nils Aall Barricelli začal simulovat genetické optimalizace a výsledky, jež publikoval v roce 1954, vešly v širokou známost [43]. V šedesátých letech pak vyšla celá řada publikací zabývajících se genetickou optimalizací a koncem šedesátých let vykrytalizoval genetický algoritmus do podoby, která se v základních rysech neliší od současné. Parametry optimalizované funkce jsou uloženy v genetickém řetězci, nad kterým jsou prováděny operace výběru, křížení a mutace. Implementaci GA v prostředí MatLab se budeme věnovat níže, neboť GA byl zvolen pro optimalizaci úloh této práce.

Evoluční strategie (ES) byla vyvíjena prakticky paralelně s GA. „V šedesátých letech byla vyvinuta na Technické univerzitě v Berlíně výzkumníky P. Bienertem, I. Rechengerem a H. P. Schwefelem.“ [9] Algoritmus je velice podobný GA, ale liší se od něj skutečností, že nevyužívá operace křížení, uplatňuje tak pouze operaci mutace a selekce. A dále, operace jsou prováděny nad reálnými čísly, nikoli nad jejich binární reprezentací, jak je tomu u GA, více [9, 43].

Genetické programování (GP) přechází od optimalizace parametrů, jak tomu je u GA, k optimalizaci struktur. Lawrence J. Fogel v roce 1964 aplikoval evoluční algoritmy k hledání konečných stavových automatů [47], Michael L. Cramer v roce 1985 položil základy moderního pojetí genetické optimalizace založené na stromové struktuře. Na uvedené práce navázal John R. Koza, který je považován za hlavního představitele v historii GP. Algoritmy GP byly implementovány zejména v prostředí programovacího jazyka Lisp [9]. Možnost evoluce symbolických struktur umožňuje syntézu celých funkcí včetně jejich struktury.



Obr. 14 Stromová reprezentace výrazu [47]

Gramatická evoluce (GE) je svým smyslem podkategorií GP, je založena na gramatice v Backus-Naurově formě (BNF, viz [48]), což umožňuje implementaci prakticky v jakémkoli programovacím jazyce, viz [9]. Gramatika sestává z terminálních a neterminálních uzlů, kdy každý neterminál je rozvinut na další terminály, případně neterminály. Příklad zápisu gramatiky [48]:

```

<syntax> ::= <rule> | <rule> <syntax>
<rule> ::= "<" <rule-name> ">" " :=" <expression>
<expression> ::= <list> | <list> "|" <expression>
<list> ::= <term> | <term> <list>
<term> ::= <literal> | "<" <rule-name> ">"
<literal> ::= "'" <text> "'" | "" <text> ""

```

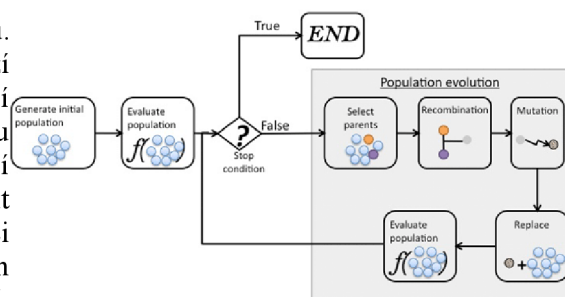
Obr. 15 Příklad obecného zápisu pravidel (zápis inspirován [48])

Diferenciální evoluce (DE) je poměrně novým typem metaheuristik spadajících do skupiny evolučních algoritmů, známa je od roku 1995, vyvinuli ji Ken Price a Rainer Storm. Podobá se genetickému algoritmu (GA), ale pracuje v reprezentaci reálných čísel (nikoli binárních jako GA) a provádí nad nimi vektorové operace, což je podobným prvkem k algoritmu SOMA. Je určena zejména k řešení numerických optimalizačních úloh. Jedním z nejvýraznějších rozdílů oproti GA je způsob křížení. [9, 43]

2.3 Realizace genetického algoritmu (GA)

Genetický algoritmus patří k nejrozšířenějším globálním řešičům optimalizačních úloh, je vhodný zejména pro úlohy, kde je hledáno optimální řešení v závislosti na velkém množství vstupních parametrů a funkční popis řešeného problému vykazuje značnou členitost prohledávané hyperplochy. GA tak pro složitější problémy překonává heuristické postupy založené na gradientních metodách, neboť gradientní metody často uvíznou v lokálním extrému. Naopak oproti složitějším meta-heuristikám (GP či GE) – nepotřebujeme-li syntetizovat celé struktury či tvary funkcí – má velkou výhodu v jednoduchosti implementace a ve vysoké efektivitě a robustnosti. Jak bylo již zmíněno, algoritmus je inspirován přírodními optimalizačními pochody, přičemž nejbližší paralelu v přírodě nalzáme u evoluce na úrovni bakterií (vývoj na vyšší organické úrovni je komplikovanější, je předmětem stálého výzkumu a je kolem něj stále mnoho otazníků). Funkce genetického algoritmu je zachycena na níže uvedené grafice.

Zaměříme se tedy krátce na paralely s přírodou. Jedinci, ať už bakterie či vyšší organismy, se nachází v prostředí, kterému se musí přizpůsobit, jestliže mají přežít. Podobně u řešení optimalizačního problému jsou hledána taková řešení, která nejlépe vyhovují podmínkám optimality. Nekvalitní řešení budou mít menší šanci uplatnit se v následující generaci (iteraci řešení problému), a to přímo nebo prostřednictvím svých potomků. Základními operacemi jsou křížení, mutace a výběr.



Obr. 16 Cyklus evolučního algoritmu [50]

K operaci křížení nalezneme nejbližší paralelu u bakterií. Jestliže dojde ke spojení dvou bakterií, jedinci si mezi sebou vymění část genetického řetězce, taková výměna může být oboustranně či jednostranně výhodná, ale může být rovněž oboustranně nevýhodná. Jestliže byla výměna nevýhodná, neznamená to, že se nestane výhodou při změně parametrů prostředí. V řešení optimalizačních problémů však oceníme zejména skutečnost, že jsou připuštěna zhoršující řešení, což zabraňuje uvážnutí v lokálním extrému. Mluvíme-li o křížení, zmíníme i různé metody výběru jedinců pro zmíněnou operaci. U bakterií je výběr více náhodný, u vyšších organismů mají lepší jedinci větší šanci vstoupit do křížení, neboť v rámci svého okolí jsou přitažlivější (viz. níže metoda typu *ruleta*), nebo své soupeře ve svém okolí poráží (viz. níže metoda typu *turnaj*).

Operace mutace bude ve své implementační podobě opět podobna více světu organismů nižších, neboť uvažujeme poměrně velkou pravděpodobnost výrazných změn. Následuje výběr jedinců pro další generaci. Zde již není přesná paralela k přírodním pochodům, kde dochází k vzniku, vývoji a zániku jedinců průběžně. Pro usnadnění se v implementaci GA omezíme na vymezení tzv. generací. Pro jednoduchost též většinou uvažujeme konstantní počet jedinců v populaci. Výběr jedinců do následující generace je rovněž stěžejní a nabízí vícero přístupů, viz níže.

Uvedme jednoduchý pseudo-kód genetického algoritmu. Níže uvedený zápis je inspirován zdrojem [50], nicméně jedná se o obecně známý algoritmus GA [9, 46].

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Pseudo-kód optimalizace pomocí GA [50]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1  P = VytvořitPočátečníPopulaci()
2  VyčísliťVhodnostJedinců(P)
3  while (PodmínkaUkončeníNesplněna)
4      P' = Křížení(P)
5      P'' = Matace(P')
6      VyčísliťVhodnostJedinců(P'')
7      VýběrProDalšíPopulaci(P' ∪ P'')
8  end

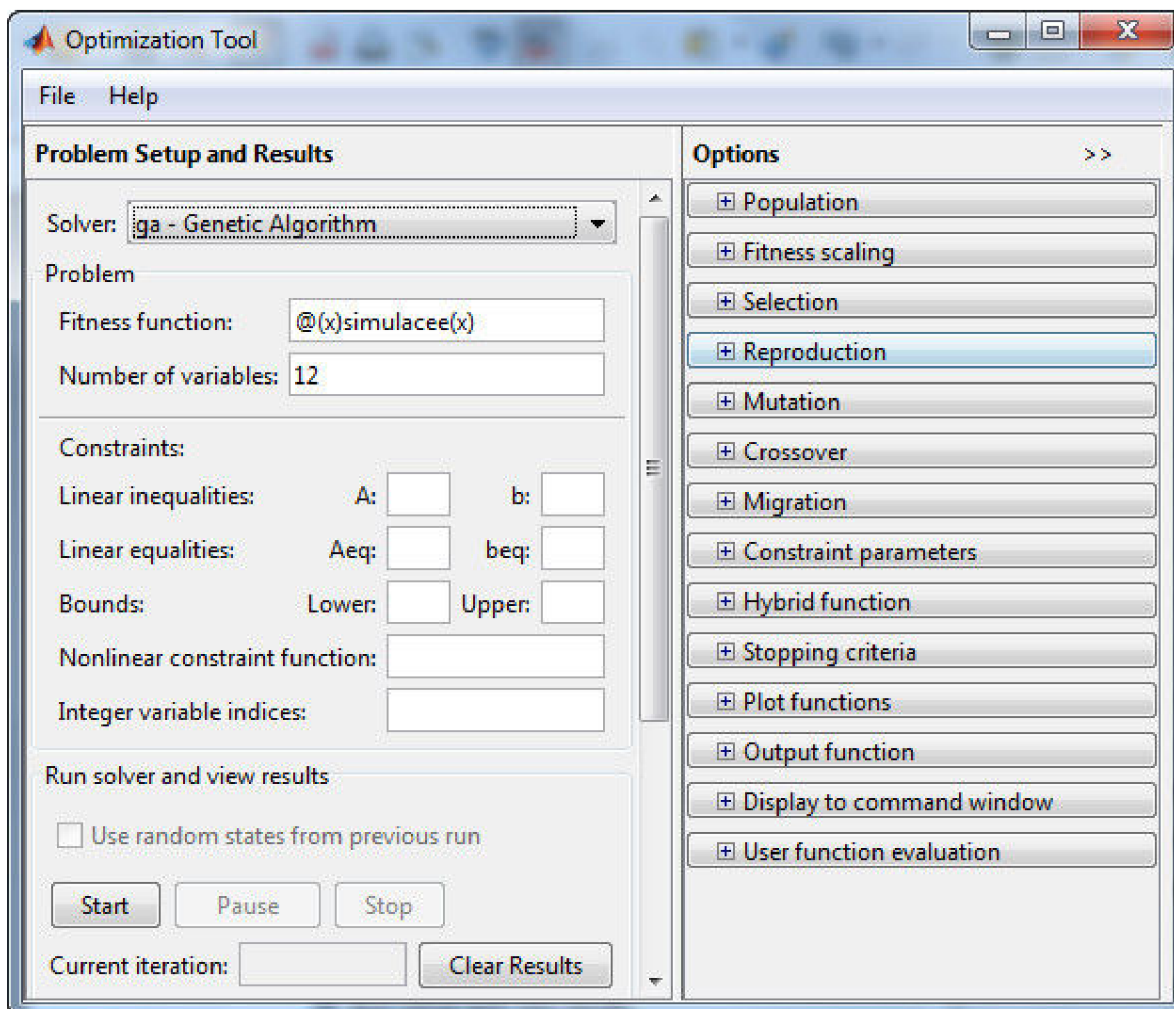
```

Uvedené schéma GA je velmi obecné a v podstatě je známo již od šedesátých let minulého století, mezi implementacemi GA se však setkáváme i s celou řadou hybridních řešení. Co se týká realizace dílčích funkcí, tj. operací výběru, křížení a mutace, je zde prostor pro velmi rozmanité přístupy, nesoucí často know how tvůrce konkrétní implementace.

Následující popis bude věnován konkrétní implementaci GA v nástroji *Global Optimization Toolbox*, dodávaném jako volitelný produkt k prostředí MatLab firmy Mathworks [54]. Níže popsané funkce a postupy jsou však natolik obecné a standardní, že je lze často vztáhnout i na alternativní produkty (existuje i mnoho volně dostupných implementací, často i ve variantě Open Source, tudíž uživatel může algoritmy dle libosti doplňovat o vlastní funkce a postupy). Nicméně, i co se týká použitého nástroje *Global Optimization Toolbox*, je produkt otevřen inovacím ze strany uživatele, neboť rozhraní umožňuje volit vlastní funkce pro všechny stěžejní operace.

2.4 Implementace GA v prostředí Matlab

Prostředí MatLab poskytuje výkonný optimalizační nástroj – *Global optimization toolbox*, který nabízí asi dvě desítky řešičů, tj. implementací různých optimalizačních nástrojů, od numerických a gradientních metod po metaheuristiky jako např. simulované žihání, ACO či GA. Nabízí se dvojitý způsob použití toolboxu, jednak lze volat samostatné funkce z příkazové řádky či z uživatelského programu, nebo se nabízí možnost použití grafického rozhraní, kterým toolbox disponuje. Použitelné je i kombinované řešení, kdy si uživatel v grafickém rozhraní nadefinuje veškeré parametry řešiče a následně exportuje veškerá nastavení do proměnné, kterou může předložit jako parametr řešiče, nebo ji uložit do souboru.

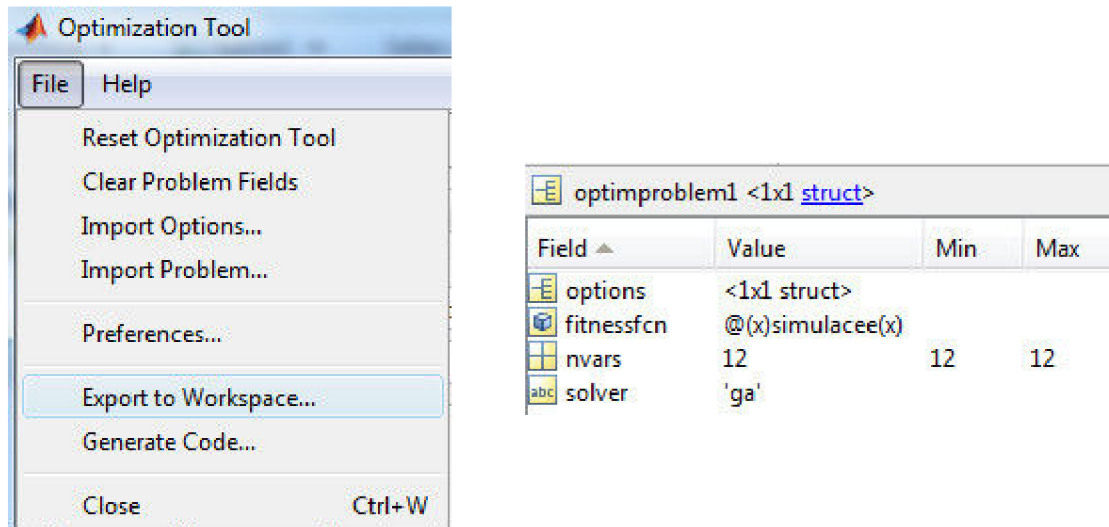


Obr. 17 Grafické rozhraní Global Optimization toolboxu

V levé horní části okna je třeba zvolit řešič a zároveň předložit řešiči optimalizovanou funkci, (ve tvaru jak je uvedeno na ilustraci). Funkce se zapíše ve tvaru, ve kterém je volána, přičemž před jméno funkce uvedeme v závorkách uvozených znakem @ proměnné, které budou předmětem optimalizace. Typicky jsou optimalizované parametry zahrnuty v rámci jedné proměnné – vektoru.

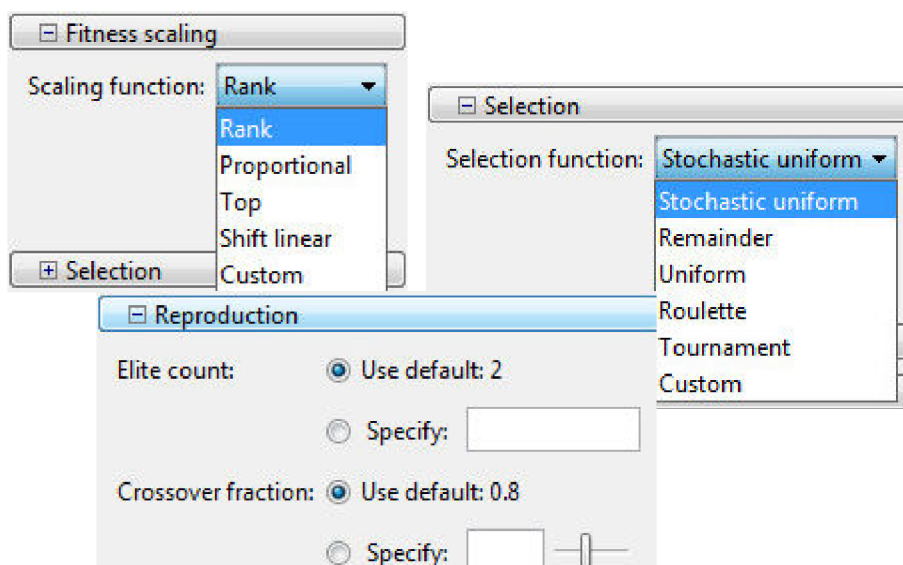
Kromě ovládacích prvků v levé části okna, nalezneme v pravé části skupinu rolovacích nabídek, kde je možno nastavovat vybrané parametry zvoleného řešiče. Mimo to, je v horní části symbol dvou šipek, kterým lze aktivovat rychlou nápovědu.

V rámci řešení složitějších problémů se jeví výhodnější variantou volání funkcí z kontextu uživatelského programu, jelikož ovládání grafických prvků by bylo pro uživatele zátěží. Přesto je vhodné potenciálního uživatele seznámit i s grafickým prostředím, kde jsou parametry řešičů přehledně uspořádány a je možné odtud snadno exportovat do prostředí celý optimalizační problém, případně pouze nastavení řešiče. Exportovaná proměnná pak obsahuje strukturu zahrnující volbu řešiče, vnořenou strukturu s parametry řešiče, dále tvar optimalizované funkce a počet optimalizovaných parametrů.



Obr. 18 Export optimalizačního problému, případně pouze nastavení řešiče do workspace a zobrazení základní struktury exportované proměnné

První skupinou parametrů GA jsou volby týkající se vlastností populace. Lze zvolit typ a velikost populace, způsob vytvoření populace, případně počáteční hodnoty. Dále se nabízí soubor funkcí pro úpravu hodnot fitness pro účely selekce. Pravděpodobnost výběru jedince je totiž určena jeho kvalitou, pokud by byly hodnoty příliš odlišné, došlo by k nadměrnému preferování kvalitních jedinců, pokud by byly hodnoty málo odlišné, neměli by kvalitní jedinci dostatečnou výhodu. Odůvodnění, popis a způsob použití funkcí je popsán v nápovědě Matlabu. Výchozím způsobem úpravy je funkce rank, jež vypočítává vhodnost pro výběr jako převrácenou hodnotu druhé odmocniny z hodnoty fitness [54].

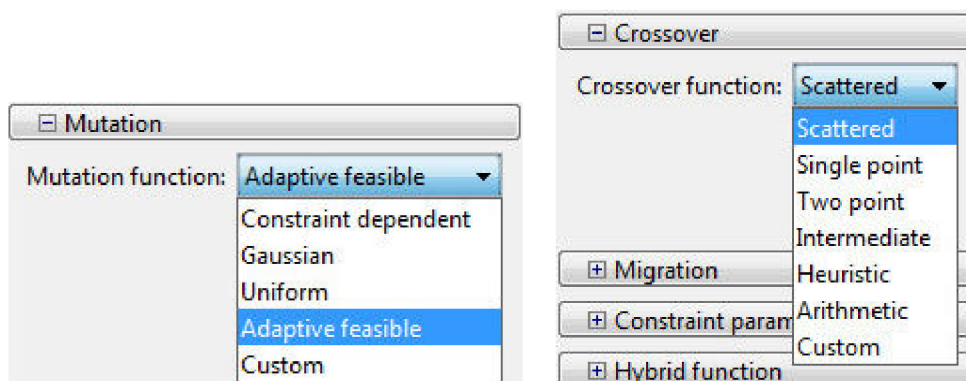


Obr. 19 Úprava vhodnosti jedinců, selekční a reprodukční pravidla

V rámci nastavení *Selection* je nabízena řada funkcí, z nichž nejvýznamnější jsou stochastická uniformní selekce, selekce typu ruleta a selekce typu turnaj. V případě stochastické uniformní selekce jsou jedinci seřazeni na intervalu dle upravené hodnoty fitness, přičemž každý jedinec zaujímá na intervalu rozlohu proporcionálně odpovídající upravené hodnotě fitness (viz *Fitness scaling*). Následně je zvolen konstantní krok jako podíl celkové délky intervalu a požadovaného počtu vybraných jedinců. Interval procházíme se zvoleným ekvidistantním krokem, výjimkou je pouze první krok, jehož délka je volena náhodně. Selekcí typu ruleta funguje obdobně, hodnoty jsou seřazeny na jednotkovém intervalu, přičemž dílčí intervaly proporcionálně korespondují s upravenou hodnotou fitness. Následně je (opakovaně dle počtu požadovaných jedinců) generováno náhodné číslo z intervalu (0; 1), pravděpodobnost výběru jedince je tedy přímo úměrná jeho vhodnosti. Selekcí turnajem pak probíhá tak, že provedeme náhodný výběr stanoveného počtu jedinců a ze vzniknuvší podmnožiny jedinců vybereme nejlepšího. Postup opakujeme. [9, 54]

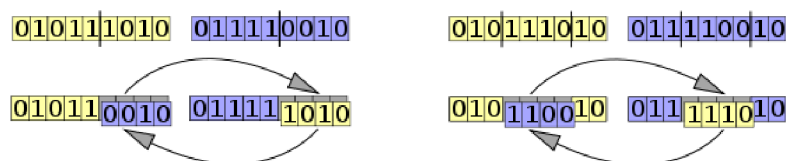
Pomocí parametrů v sekci *Reproduction* stanovíme počet elitních jedinců, tzn. počtu n nejkvalitnějších řešení, která budou (bez modifikací) použita v následující generaci. Dále stanovíme, jak velká část nové populace vznikne křížením. Zbylí jedinci budou podrobeni pouze operaci mutace.

V sekci *Mutation* volíme typ mutace včetně případných parametrů. Výchozím typem je Gaussovská mutace, což znamená, že k jednotlivým prvkům vektorů reprezentujících jedince jsou přičítána náhodná čísla, jejichž pravděpodobnost je dána gaussovskou křivkou, jejíž parametry je možné nastavit. Pokud jsou nastavena omezení, je výchozí funkcí adaptivní mutace, více [54]. Mutace je nezbytným prvkem GA, neboť zabráňuje degradaci populace na málo odlišná řešení [9].



Obr. 20 Parametry mutace a selekce

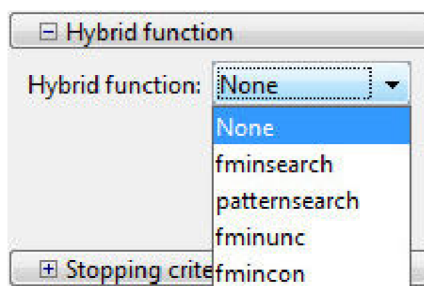
Křížení je klíčovým prvkem GA. Určuje způsob, kterým budou vytvářena nová řešení pro následující generaci. Jedná se o bod, který bývá často doplňován o nové myšlenky a postupy. Stejně jako u předešlých operací, je i zde možno volit vlastní uživatelskou funkci pro definici způsobu křížení. Nicméně nabídka předpřipravených funkcí je dostatečně pestrá. Všechny uvedené funkce jsou zdokumentovány a na webu bychom našli řadu alternativ aplikovatelných nejen v Matlabu. Následující obrázek ilustruje nejjednodušší formy křížení, tj. jednobodové a dvoubodové křížení nad binárním řetězcem reprezentujícím genotyp jedince.



Obr. 21 Jednobodové a dvoubodové křížení [49]

Další nastavení jako jsou funkce pro vykreslení výsledků v této části nebudeme rozebírat, neboť jejich použití je do značné míry intuitivní a u komplikovanějších problémů předpokládá vytvoření vlastních zobrazovacích funkcí.

Za zmínku ještě stojí možnost využití tzv. hybridních funkcí, tj. řešičů, které budou výsledné řešení (nalezené pomocí GA) dále optimalizovat. Jak bylo řečeno, GA překonává klasické gradientní metody nejvýrazněji v případě optimalizace velmi členitých funkcí závislých na mnoha neznámých. Nalezené řešení pomocí GA však nemusí být ještě tím nejlepším a v jeho okolí se může nacházet řešení lepší, přičemž je-li funkce v dané oblasti „ukázněná“, může klasická gradientní metoda přinést velký užitek.



Obr. 22 Nabídka hybridních funkcí

3 VÍCE-KRITERIÁLNÍ OPTIMALIZACE A PARETOVO OPTIMUM

Problematika optimalizace proniká nejen napříč všemi technickými obory, ale je součástí našeho každodenního života, neboť každodenně jsme – ať už vědomě či podvědomě – nuceni dělat mnohá rozhodnutí, a tak v podstatě řešit optimalizační úlohy (ať už se jedná o výběr nového televizoru či výběr jídla v restauraci). Proces rozhodování je ve většině případů determinován více hledisky (cena, kvalita, dostupnost, atd.) přičemž jednotlivým faktorům přidáváme různou (subjektivní) váhu.

Agregace vícero různých kritérií do jedné kritériální (hodnotící) funkce může být dosažena váhováním dílčích kritérií. Nicméně nastavení vah dílčích kritérií je samo o sobě optimalizační úlohou. V mnoha případech tak vytváříme vzorce ovlivněné subjektivním vnímáním, či vzorce méně obecné.

Mějme tedy nejprve jednokritériální optimalizační problém popsany jednou kritériální funkcí. Zmíněná oblast optimalizace se v anglické literatuře nazývá povětšinou *single-object optimization*, přičemž optimalizační problém je formulován následovně:

$$\mathbf{x}_{opt} = \underset{\mathbf{x} \in X^m}{\operatorname{arg\,opt}} f(\mathbf{x}) \quad (5)$$

kde f je skalární funkce a X je m -dimenzionální prostor přijatelných řešení (*feasible region*, *feasible space*). [9, 22]

Argument optima – optimální vektor vstupních parametrů – je determinován optimem v hodnotě pro kritériální funkci, která je v případě jedno-kritériální optimalizace funkcí skalární. Optimálních řešení tak může existovat více, dokonce i nekonečně mnoho, ale všechny musí vést na stejnou hodnotu kritériální funkce, tj. na globální minimum (resp. maximum, dle formulace úlohy).

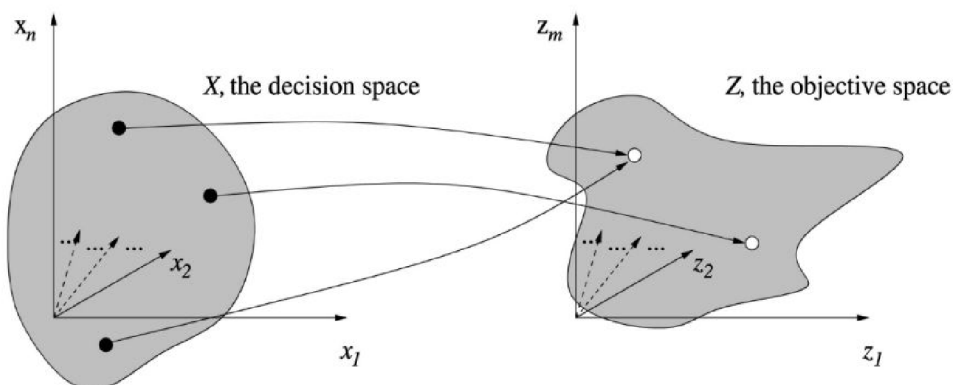
Přístupme nyní k pojmu vícekritériální (nebo též multikritériální) optimalizace, v anglické literatuře ji najdeme pod termíny *multiobjective optimisation*, *multicriteria optimisation*, nebo *pareto optimisation*. V případě pojmu pareto-optimalizace se ale jedná o konkrétnější termín.

Výsledek multikritériální optimalizace již není skalár, ale vektor, jehož složky odpovídají hodnotám dílčích skalárních jednokritériálních funkcí. Problém lze matematicky formulovat následovně:

$$\mathbf{x}_{opt} = \underset{\mathbf{x} \in X^m}{\operatorname{arg\,opt}} [f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), \dots, f_n(\mathbf{x})] \quad (6)$$

kde $n > 1$ a X je m -dimenzionální prostor přijatelných řešení (více [9, 22]).

V případě jedno-kritériální optimalizace byla za optimální prohlášena ta řešení, pro něž nabývala kritériální funkce svého globálního extrému v hodnotě. V případě multikritériální optimalizace je ale situace složitější. Výslednou hodnotou kritériální funkce již není skalár, ale vektor hodnot, jehož složky odpovídají výsledkům dílčích kritériálních funkcí. Uvažujeme-li, že alespoň dvě parciální kritéria jsou vzájemně v *konfliktu*, nastane situace, kdy zlepšení jedné složky kritériální funkce způsobí u jiné složky zhoršení. Nebude tedy již existovat jediné optimální řešení, ale celá množina navzájem nedominantních řešení, z nichž každé má nárok být nazváno optimálním řešením dané úlohy.

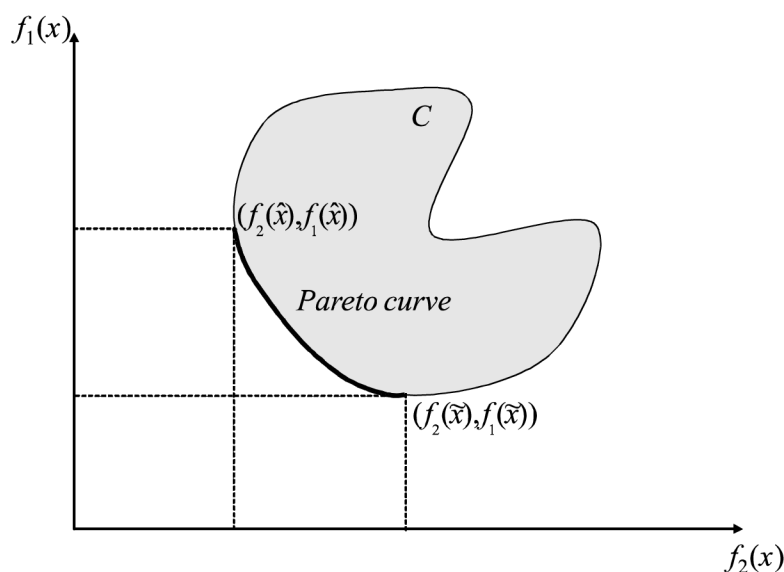


Obr. 23 Zobrazení optimalizačního problému o n vstupních parametrech do m -dimenzionálního prostoru řešení kriteriální funkce [52]

Nyní je na místě zavést pojem *pareto-optimální řešení*. Optimalizace ve smyslu hledání pareto-optimálních řešení je chápána více jako koncept než definice. Zavedl ji v roce 1906 italský inženýr, sociolog a ekonom Vilfredo Federico Damaso Pareto [23, 24]. Definice zní [24]:

Paretovo optimum: Bod $x_{opt} \in X$ je pareto-optimální, jestliže neexistuje žádný bod $x \in X$, pro který platí, že $f(x) \leq f(x_{opt})$, a $f_i(x) < f_i(x_{opt})$ pro alespoň jednu funkci.

Pokusíme-li se reinterpretovat výše uvedenou větu, pareto-optimálními řešeními můžeme nazvat ta, pro která platí, že nelze dosáhnout zlepšení u kterékoli dílčí hodnotící funkce, aniž by vedlejším efektem bylo zhoršení u jiné dílčí kriteriální funkce. Pareto-optimální řešení tvoří v prostoru řešení množinu diskrétních bodů, spojitou či po částech spojitou křivku či hyperplochu. V případě dvoudimenzionálního multikriteriálního problému bude množina pareto-optimálních řešení, tedy vzájemně nedominantních optimálních řešení úlohy, ležet na *Paretově křivce*, alternativně může být použit i pojem *Paretova hranice*.



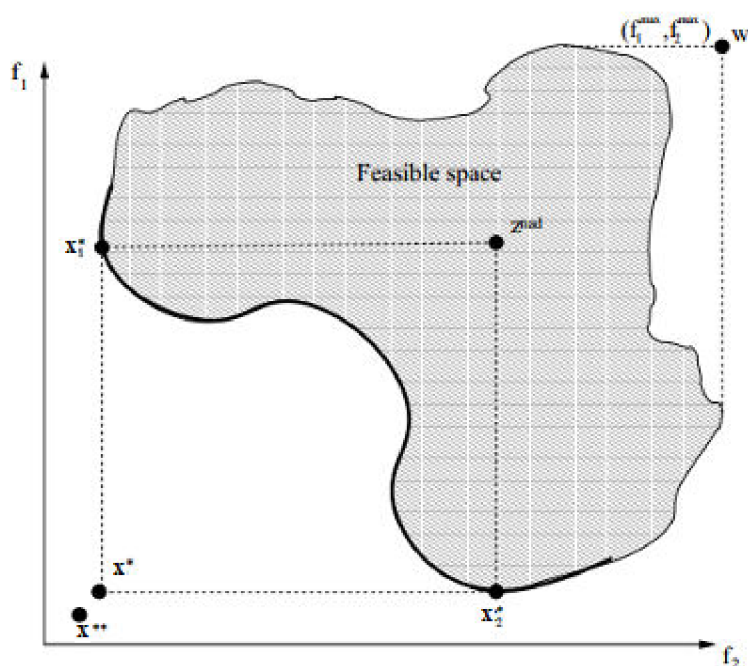
Obr. 24 Paretova hranice [22]

Paretova hranice je v problematice vícekriteriální optimalizace evidentně prioritním zájmem našeho zkoumání. Přesto si uvedme ještě několik pojmů, se kterými se v zmiňované problematice potkáváme. Prvním pojmem bude *ideální vektor (řešení)*, jedná se o imaginární řešení, jehož složky odpovídají optimálním řešením dílčích částí kriteriální funkce (viz ilustrační obrázek).

$$\mathbf{x}_{opt}^x = [x_{1,opt}^x, x_{2,opt}^x, x_{3,opt}^x, \dots, x_{n,opt}^x] \quad (7)$$

Druhým pojmem je *utopický vektor*, pro který platí, že jeho složky jsou o trochu lepší než složky ideálního vektoru.

$$\begin{aligned} \mathbf{x}_{opt}^{xx} &= [x_{1,opt}^{xx}, x_{2,opt}^{xx}, x_{3,opt}^{xx}, \dots, x_{n,opt}^{xx}] \\ x_{i,opt}^{xx} &< x_{i,opt}^x - \epsilon; \quad \epsilon > 0, \quad i = 1, 2, \dots, n \end{aligned} \quad (8)$$



Obr. 25 Ideální vektor x^* , utopický vektor x^{**} , nadir w [52]

4 TESTOVACÍ SOUSTAVY

Níže představené metaheuristické postupy optimalizace regulačních soustav se osvědčily jako velmi výkonný nástroj pro hledání optimálního tvaru a parametrů obecných lineárních regulátorů. Aby však bylo možné doložit kvalitu optimalizačního procesu, je zapotřebí výsledky srovnat s výsledky jiných metod (ať už klasických či heuristických). Kromě alternativních postupů, které budou nastíněny dále, je zapotřebí vybrat reprezentativní soustavy, na nichž budou optimalizační postupy aplikovány a následně porovnány.

Pro uvedené účely bylo vybráno několik reprezentativních (lineárních) soustav, z nichž některé jsou dále doplněny rovněž o základní nelinearity (saturace, mrtvá zóna, atd.). Mohla by vyvstat otázka, proč se zaměřujeme právě na regulaci lineárních soustav. Reálné soustavy jsou často nelineární (prakticky všude dříve či později narazíme na saturaci, nemluvě o dalších parazitních nelinearitách), nicméně v okolí pracovního bodu lze soustavy obvykle velmi dobře linearizovat a následná práce s lineárními modely je značně jednodušší a tudíž preferovaná. S ohledem na možnou změnu parametrů soustavy (v přiměřeném rozsahu) je nutno brát zřetel rovněž na robustnost nalezeného řešení, právě vůči změně parametrů soustavy. Omezíme-li se tedy na soustavy lineární (případně doplněné o dopravní zpoždění a saturaci), pak se můžeme zaměřit na vybrané základní tvary přenosů.

Vedle kmitavých soustav, pro něž budeme mít rovněž jednoho zástupce, se většina soustav chová jako systém n -tého řádu s dolnopropustním charakterem. Většinu takových systémů pak lze aproximovat jedním z následujících tvarů, (viz přenosové funkce) [4]:

1. soustava 1. řádu s dopravním zpožděním	$G(s) = \frac{k}{T s + 1} e^{-T_d s}$
2. soustava druhého řádu	$G(s) = \frac{k}{(T_1 s + 1)(T_2 s + 1)}$
3. soustava druhého řádu s dopravním zpožděním	$G(s) = \frac{k}{(T_1 s + 1)(T_2 s + 1)} e^{-T_d s}$
4. soustava n -tého řádu se stejnými časovými konstantami	$G_p(s) = \frac{k}{(T s + 1)^n}$

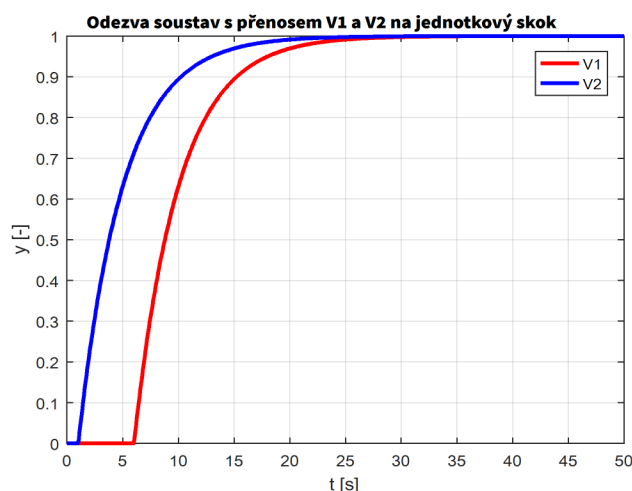
Z uvedených základních tvarů byla vybrána soustava prvního řádu s dopravním zpožděním. Na první pohled by se mohlo zdát, že se jedná o nejjednodušší variantu – to možná v některých případech ano – nicméně přítomnost dopravního zpoždění regulaci velice komplikuje, zejména pokud je dopravní zpoždění srovnatelné, nebo dokonce větší než dominantní (v tomto případě jediná) časová konstanta systému.

Systém prvního řádu s dopravním zpožděním je zde prezentován ve dvou variantách, z nichž varianta V_1 disponuje dopravním zpožděním, které převyšuje svoji hodnotou velikost časové konstanty systému, kdežto v případě varianty V_2 , je dopravní zpoždění zvoleno menší než časová konstanta soustavy. Časové konstanty a dopravní zpoždění byly vybrány nikoli náhodně, ale podle publikace [30], pro lepší možnosti porovnání výsledků s klasickými metodami návrhu parametrů 1DOF PID a 2DOF PID.

Soustavy V_1 a V_2 jsou popsány následujícími operátorovými přenosy:

$$V_1(s) = \frac{1}{4s + 1} e^{-6s} \quad V_2(s) = \frac{1}{4s + 1} e^{-1s} \quad (9)(10)$$

Následující graf vyobrazuje přechodové charakteristiky obou vybraných soustav.



Obr. 26 Přechodová charakteristika pro soustavy V_1 a V_2

Bylo by možné dále vybírat ze základních tvarů soustav, další varianty však vybereme ze soustav řádu vyššího (v našem případě třetího), zejména proto, že u nich prakticky vždy lze aplikovat návrh parametrů PID metodou dle Ziegler-Nicholse či pomocí dalších běžně používaných standardních metod syntézy regulátorů. Uvedené soustavy třetího řádu jsou pracovně označeny jako F_1 a F_2 . Verze označená F_1 je setrvačnou nekmitavou soustavu, kterou je možno modelovat sériovým spojením tří setrvačných článků. Soustavu je možno popsat operátorovým přenosem:

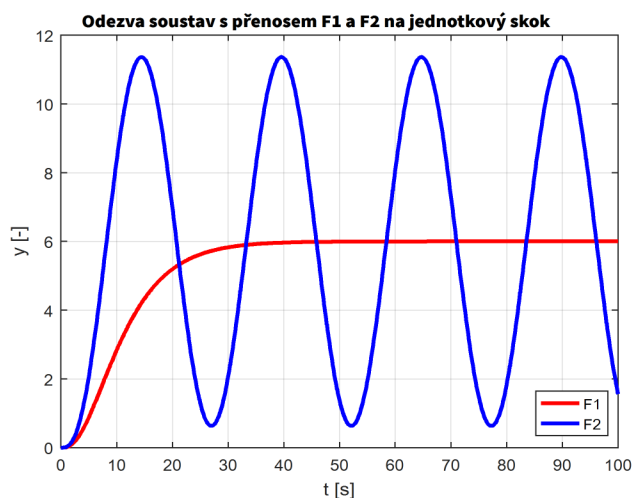
$$F_1(s) = \frac{6}{48s^3 + 44s^2 + 12s + 1} = \frac{0,125}{(s+0,5)(s+0,25)(s+0,1667)} \quad (11)$$

Druhá varianta soustavy vyššího řádu je označena F_2 a její operátorový přenos je:

$$F_2(s) = \frac{6}{32s^3 + 16s^2 + 2s + 1} = \frac{0,1875}{(s+0,5)(s+0,25i)(s-0,25i)} \quad (12)$$

Jedná se o soustavu na mezi stability – kmitavou.

Přechodové charakteristiky obou uvedených soustav jsou uvedeny v grafu:



Obr. 27 Přechodová charakteristika pro soustavy F_1 a F_2

5 VYUŽITÍ EVT PŘI NÁVRHU REGULÁTORŮ

Pro návrh regulátorů resp. pro optimalizaci jejich parametrů byla během historie automatického řízení vyvinuta celá řada algebraických, grafických a empirických metod, a to zejména v průběhu minulého století. Mezi grafické metody patří například návrh tvaru regulátoru v oblasti frekvenčních charakteristik [4]. Mezi algebraické metody uveďme metodu optimálního modulu [4], návrh regulátorů s využitím polynomiální teorie řízení [17] či vybraná integrální kritéria (kvadratické integrální kritérium) [4]. Mezi empirické metody zařadíme například metodu Ziegler-Nichols pro seřízení PID regulátoru [11].

Všechny uvedené metody byly navrženy pro optimalizaci vybraných parametrů vybraných typů regulátorů, přitom nejsou ani zdaleka schopny pojmut všechny kvalitativní vlastnosti soustavy. Jestliže začneme jednotlivé požadavky kombinovat, stává se analytický výpočet neúnosně náročný, ve většině případů obtížně proveditelný či zcela neproveditelný. Zejména začnou-li v systému působit dopravní zpoždění a nelinearity [5]. Ostatně i některá zdánlivě jednoduchá optimalizační kritéria, neumožňují obecný analytický výpočet a vyžadují numerické řešení (například integrální kritérium ITAE, ITSE, atd.).

Mnohdy se spokojíme s aproximacemi a zjednodušováním modelů reálných soustav, avšak ne vždy je aproximace dostatečně vystihující skutečné parametry, či není dostatečně redukcující a výpočet ulehčí jen málo. Nezbyvá tedy než hledat řešení jinak. Pro množství kombinací nastavení hodnot jednotlivých parametrů regulátoru není možné projít všechna řešení a je nutné optimální řešení nalézt užitím vhodné ne-analytické optimalizační metody. Klasické numerické gradientní metody však často selhávají či nepodávají dostatečně kvalitní výsledky, neboť prohledávaný prostor možných řešení je mnoho-dimenzionální, velmi členitý a obsahuje velké množství lokálních extrémů. Je proto zapotřebí využít vhodných metaheuristik pro hledání kýženého globálního extrému řešené úlohy.

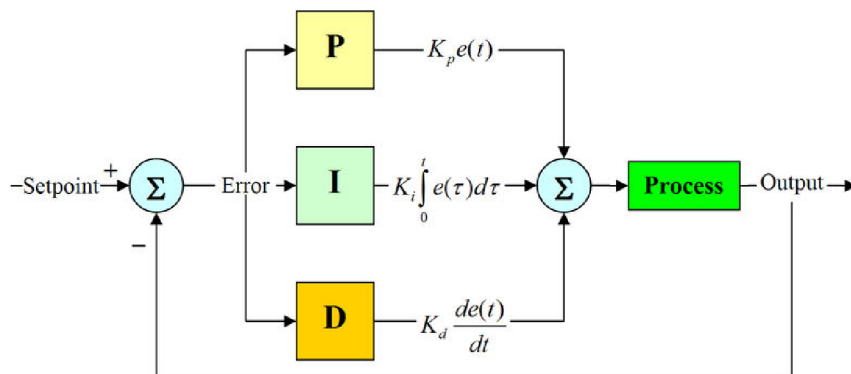
Otázku, zda mají evoluční výpočetní techniky (EVT) místo v oblasti návrhu a optimalizace regulačních obvodů, lze zcela jistě považovat za otázku řečnickou. Již řadu let se využívá nejrůznějších metaheuristik (založených na EVT) k optimalizaci základních regulátorů, kdy hledanými parametry jsou například koeficienty jednotlivých složek PID regulátoru, případně filtrační složka či perioda vzorkování (v případě diskrétní varianty PID). V současné době lze pozorovat stálý nárůst aplikací EVT v regulační technice. Pro vysokou setrvačnost a konzervativnost oboru (což není špatně) nejsou dosud EVT nasazovány v masovém měřítku, nicméně lze očekávat pokračování nárůstu aplikací využívajících právě EVT. Zdůrazněme, že pro úlohy, kde si vystačíme s klasickými metodami, nemá smysl hledat jiné metody řešení, protože klasické metody [4, 7, 11] vedoucí na exaktní vyjádření řešení nabízí výhodu určit jednoznačné řešení. Na druhou stranu jsou možnosti klasických metod omezené.

Evoluční výpočetní techniky většinou nepostihnou požadované nastavení regulátorů analyticky, ale naleznou je sofistikovaným prohledáváním prostoru možných řešení. Zmíněná myšlenka je zároveň nosnou myšlenkou této práce.

5.1 Optimalizace parametrů PID s využitím GA

5.1.1 Koncepce a realizace

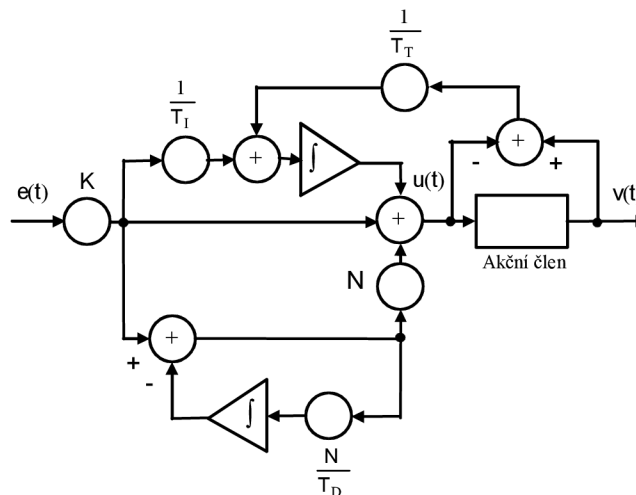
Regulátory typu PID byly zmíněny již v kapitole 2.4, kde je mimo jiné uvedeno, že se jedná o bezesporu nejrozšířenější regulátory v automatizaci. Mezi odborníky se dokonce vyskytují názory, že lepší univerzální regulátor než PID nalézt nelze. Popis PID je většinou uváděn v jeho spojitě formě, i když v současné době je velké množství regulátorů realizováno pomocí číslicových obvodů. Nicméně číslicové obvody dosahují natolik vysokých vzorkovacích frekvencí, že lze realizovat regulátory defacto spojitě, uvážíme-li výrazně vyšší časové konstanty řízených soustav.



Obr. 28 Regulační obvod s ideálním regulátorem PID [51]

Výše uvedený obrázek znázorňuje schematicky zapojení zpětnovazebního regulačního obvodu s ideálním regulátorem PID. Uvažujeme-li elektronické regulátory, které jsou nejrozšířenější, můžeme pozorovat, že složky P (proporcionální), I (integrační) a D (derivační), jsou realizovány jako jednotlivé členy (ať už hardwarově či softwarově). Popis regulátoru PID známe od minulého století, nicméně našli bychom regulátory odpovídající tvaru PI i v podstatně hlubší historii, ale jelikož se jednalo povětšinou o mechanické regulátory, oddělení jednotlivých složek (byť pomyslně) by bylo obtížné.

Nutno podotknout že výše uvedený model PID regulátoru je pouze teoretický, jelikož obsahuje ideální derivační člen. Při skokové změně regulační odchylky by derivační člen vystavil na svém výstupu (ideální) Diracův impuls, což není fyzikálně možné. Zároveň by byl ideální derivační člen příliš citlivý na (byť slabý) vysokofrekvenční šum, kterým bývá reálný obvod zatížen. Pro derivační složku je proto aplikována *filtrace*. Filtrační konstanta N se běžně se volí v rozsahu 3 – 20.

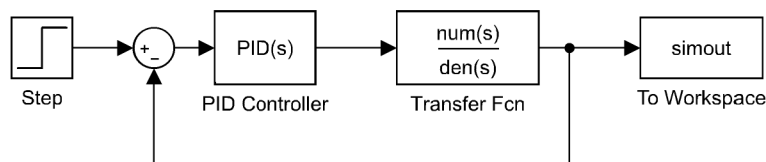


Obr. 29 Regulační obvod s filtrace derivace a omezením přebuzení [14]

5.1.2 Optimalizace PID s využitím GA

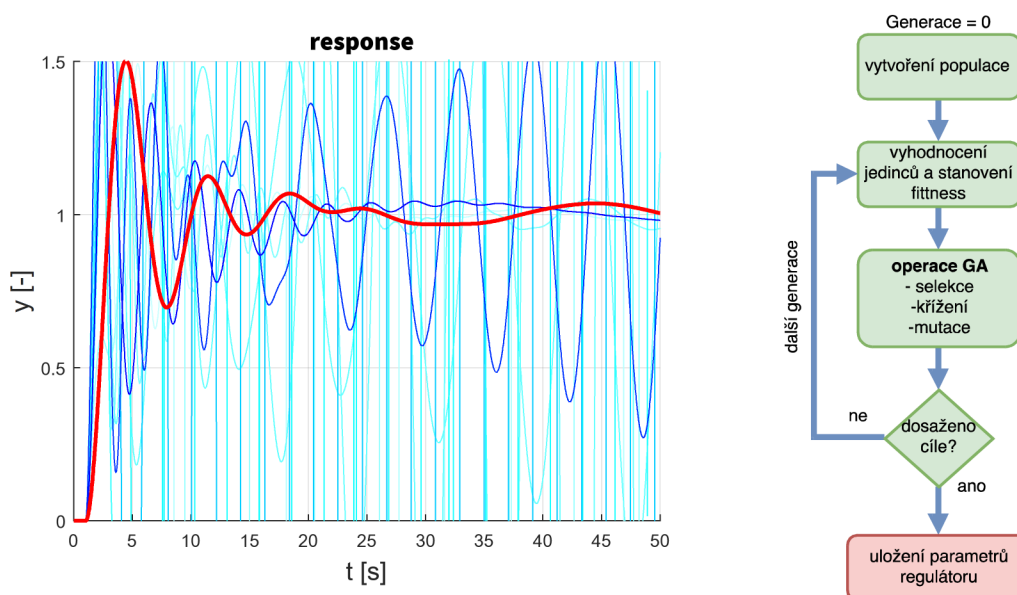
Optimalizační metoda je založena na simulaci. Genetický algoritmus vytváří jednotlivé kombinace parametrů PID, tyto kombinace nahlížíme jako fenotyp jedince v populaci řešení.

Na schématu je uvedeno zjednodušené schéma regulačního obvodu, jedná se zpětnovazební regulační smyčku obsahující PID regulátor (s filtrací derivační složky) a regulovanou soustavu. Parametry regulátoru jsou nastavovány genetickým algoritmem. Výsledky simulace (časové průběhy) jsou exportovány do workspace Matlabu kde jsou dále zpracovány (je vypočítána kvalita jedince, tj. ukazatele kvality).



Obr. 30 Zjednodušené schéma simulačního modelu

Genetický algoritmus prochází jednotlivými iteracemi dokud není dosaženo požadovaných kvalit nebo dokud není dosaženo maximálního počtu iterací. Hlavní cyklus zahrnuje vyhodnocení vhodnosti jedinců (tj. kombinací parametrů PID), selekci, křížení a mutaci a výběr pro následující generaci. Výsledkem optimalizace jsou optimální parametry PID vzhledem k nastavenému kritériu optimality.



Obr. 31 Ilustrační obrázek procesu hledání optima kritériální funkce pomocí GA

5.2 Pareto-optimalizace parametrů PID s využitím GA

5.2.1 Motivace

Jedním z přístupů k hodnocení kvality regulátorů při multikriteriálních požadavcích na optimalizaci je aplikace pareto-optimalizace. V kapitole *Kvalita regulace* byla uvedena celá řada vlastností ovlivňujících kvalitu nalezeného řešení. Jelikož byla již v dřívější době vnímána potřeba hledání kompromisů (při optimalizaci jednotlivých parametrů), vznikla pokročilejší kritéria optimality jako například zmíněná integrální kritéria, která primárně minimalizují integrální plochu regulační odchylky, vedlejším efektem je pak skutečnost, že je nalezeno řešení, které je dobrým kompromisem v optimalizaci jednotlivých základních kritérií, jako je například maximalizace rychlosti přeběhu a ustálení a minimalizace překmitu.

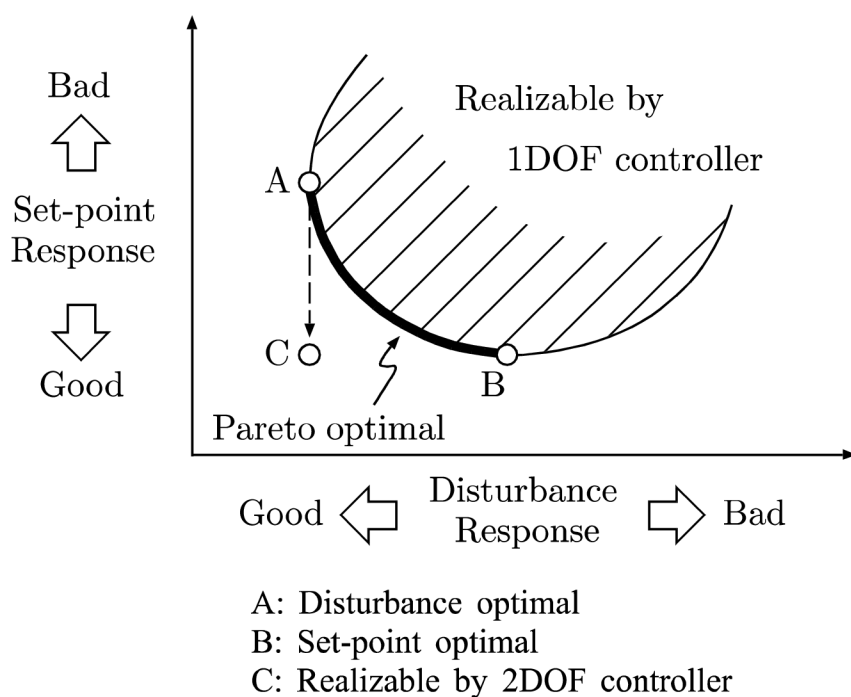
Avšak ne vždy je poměr vah, se kterými byl učiněn kompromis, vyhovující. Často je žádoucí klást větší či menší důraz na jednotlivá dílčí kritéria. A pokud zvýšíme své nároky: Kde je kritérium, které by zahrnuje optimalizaci dalších vlastností, jako například robustnost systému vůči působící poruše, omezení kmitavosti akčního zásahu, energetická efektivita akčního zásahu, atd.? V tomto případě nemáme žádné univerzální kritérium a je nutné znovu přejít k optimalizaci na základě více kritérií, což je bezesporu výzva pro oblast evolučních výpočetních technik.

Na první pohled může multikriteriální optimalizace působit nedůvěryhodně. Jakým způsobem máme posuzovat optimalitu na základě kritérií, která spolu jen slabě nesouvisí? Jakým způsobem dávat do porovnání výsledek funkce která minimalizuje kvadratickou plochu regulační odchylky a funkci, která minimalizuje zvlnění akčního zásahu?

Naším úkolem však není provést normalizaci jednotlivých kritérií a ta společně poměřovat. Nastoleným úkolem je najít taková řešení, která budou pareto-optimální. Zatímco dříve jsme se zaměřili na minimalizaci integrální plochy regulační odchylky, nyní budeme hledat obdobně kvalitní řešení (třeba i o málo horší), pro která bude ale dosaženo zlepšení pro jiný parametr určující kvalitu regulace.

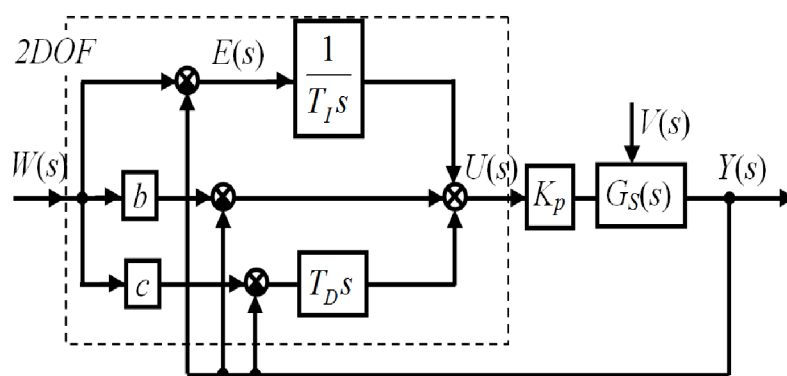
5.2.2 Optimalizace regulátorů PID s jedním a dvěma stupni volnosti

Pareto-optimalizace regulačních úloh je jedním z velmi zajímavých a perspektivních přístupů. Uvedeným tématem se zabývá řada publikací. Za mimořádně zajímavou a kvalitní považují práci autorů Mituhiko Araki a Hidefumi Taguchi prezentovanou ve článku *Two degree of freedom PID Controllers* [2]. Primární zde není multikriteriální optimalizace jako taková, ale autoři diskutují vlastnosti 1DOF PID a 2DOF PID regulátorů z pohledu optimalizace vlastností při přenosu řízení a přenosu poruchy. Autoři zde uvádějí následující ilustraci: *Obr. Paretova hranice pro obecný systém řízený 1DOF/2DOF PID* [2]



Obr. 32 Ilustrativní příklad pareto-optimizace regulátoru [2]

Jsou zde hodnocena dvě kritéria, jednak odezva na řízení a odezva na poruchu. Obě uvedené funkční hodnoty je cílem minimalizovat. Zároveň autoři uvádějí, že pomocí aplikace 2DOF PID je možné prolomit Parotovu hranici klasického PID a dostat se na lepší řešení. Uvedený obrázek je však pouze obecný, vhodný pro ilustraci problému, je však nutno uvést, že pro každý systém bude průběh pareto-hrance PID regulátoru vypadat jinak (viz kapitola 6.2.3). A dále, že ani 2DOF PID často nebude mít jediné globálně optimální řešení, ale že i pro 2DOF PID bude existovat pareto-hranice.

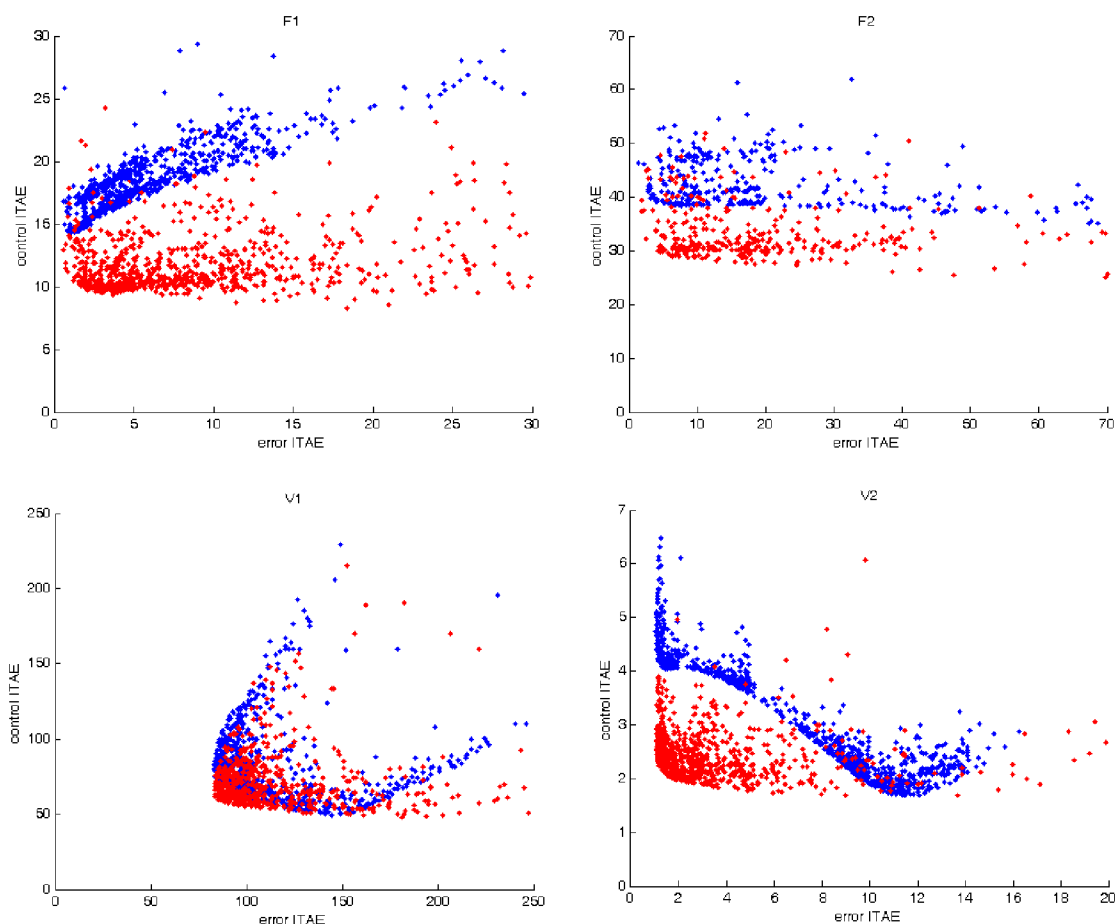


Obr. 33 PID regulátor ve variantě 2DOF

5.2.3 Výsledky testování

Za účelem otestování a zhodnocení teoretických závěrů bylo již provedeno mnoho testů. Vycházíme ze stejného pojetí dvou-kriteriálního posuzování (jako autoři [2]), a sice, že budeme posuzovat kvalitu odezvy na řízení a na poruchu. Jako hodnotící funkce bylo zvoleno integrální kritérium ITAE pro hodnocení kvality přechodového děje při působení řízení a rovněž kritérium ITAE pro hodnocení přechodového děje při působení poruchové veličiny. V obou případech se jednalo o odezvu na jednotkový skok (řízení resp. poruchy). Mohlo by být sice namítnuto, že uvedené hodnoty nelze srovnávat, neprovedeme-li minimálně normalizaci. Znovu zdůrazňeme, že není účelem srovnávat jednotlivé vlastnosti ovlivňující kvalitu, nýbrž hledat paretovska optima.

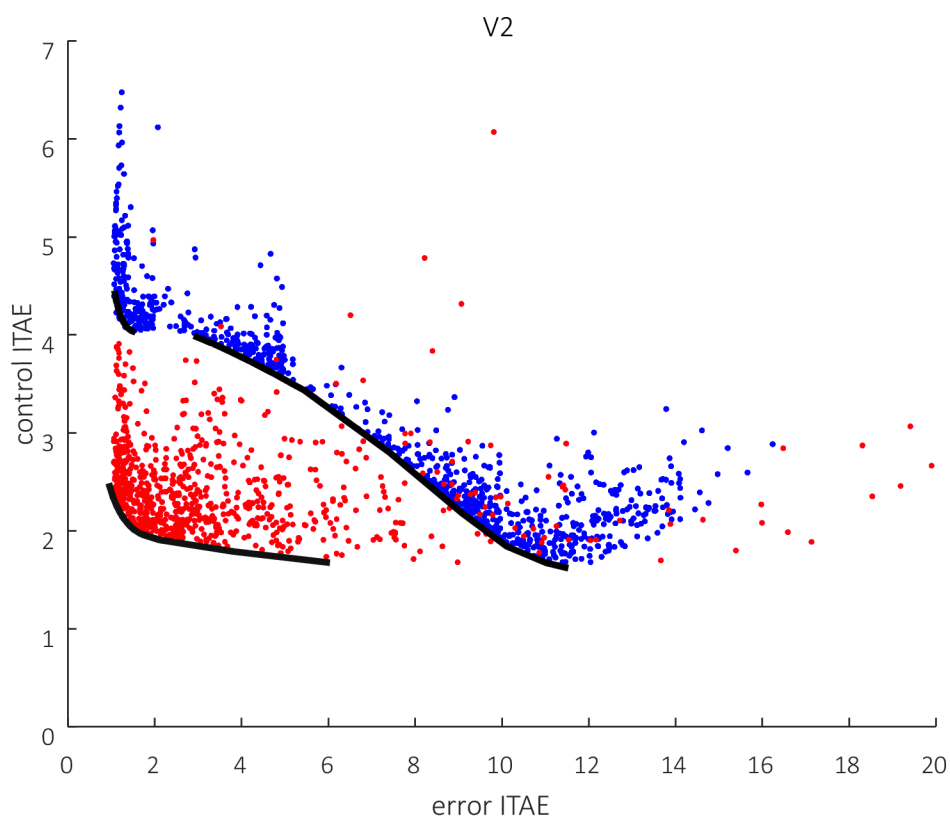
Uvedme výsledky optimalizace PID a 2DOF PID pro modelové soustavy V_1, V_2, F_1 a F_2 :



Obr. 34-37 Výsledky multikriteriální optimalizace pro soustavy o přenosech F_1, F_2, V_1, V_2

Bylo prakticky ověřeno, že 2DOF regulátor dosahuje mnohem lepších pareto-optimálních řešení než klasická PID regulátor. Ale skutečně pouze z pohledu pareto-optimalizace, pokud bychom se zaměřili vždy pouze na jedno kritérium, pak dosáhneme (ve většině případů) podobných výsledků s oběma regulátory. Může teoreticky nastat, že i v jednom samostatném kritériu bude 2DOF PID vykazovat lepší výsledky, opačně tomu být nemůže, neboť 2DOF PID v nejhorším případě degraduje v klasické PID.

I tabulka uvedená níže zachycuje skutečnost, že pokud nebudeme uvažovat omezení vlivu poruchy, pak se s oběma typy regulátorů dostaneme na podobné výsledky, viz poslední řádky tabulky. Ale pokud stanovíme omezující podmínky na hodnotu některého z kritérií přísněji, pak v mnoha případech převyšuje 2DOF PID svojí kvalitou klasické PID.

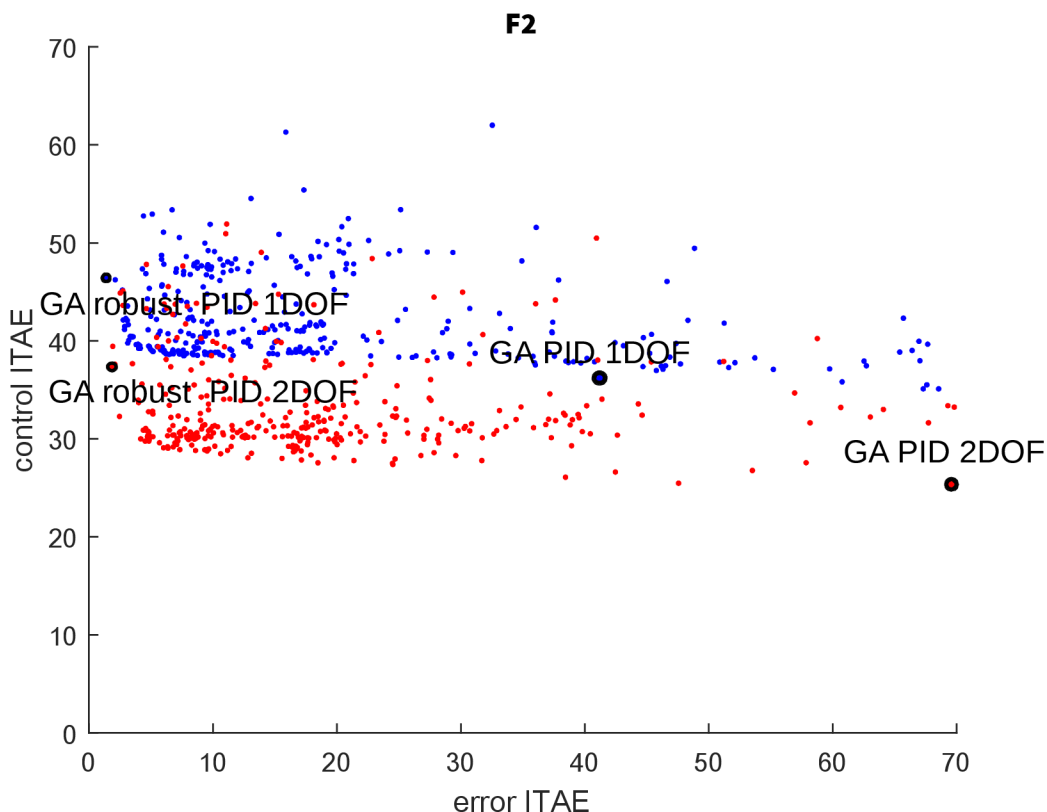


Obr. 38 Výsledky multikriteriální optimalizace pro soustavu o přenosu V_2

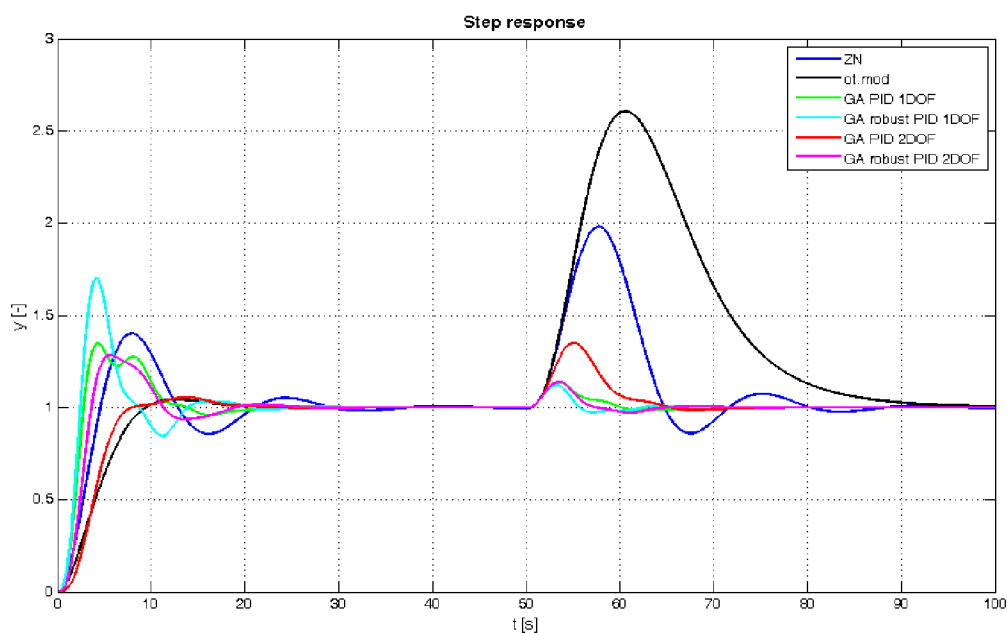
Omezující podmínka: maximální hodnota ITAE pro skok poruchy	ITAE skoku řízení pro nejlepší nalezené řešení	
	1DOF	2DOF
1,06	4,39	2,70
1,25	4,16	2,15
1,50	4,06	2,06
1,75	4,06	2,02
2,00	4,06	2,01
2,50	4,04	1,92
3,00	4,02	1,84
4,00	3,80	1,80
5,00	3,59	1,77
6,00	3,27	1,74
7,00	2,92	1,73
8,00	2,62	1,71
9,00	2,21	1,68
10,00	1,86	1,68
11,00	1,71	1,68
12,00	1,68	1,68
>12,00	1,68	1,68

Tabulka 3: Srovnání výsledků kvality regulace soustavy V_2 pomocí 1DOF PID a 2DOF PID, v levém sloupci je uvedeno omezení stanovené jako maximální dovolená integrální plocha odchylky při působení poruchy

Podívejme se ještě na jednu ze soustav. Pro soustavu F2 nebudeme srovnávat výsledky tabulkou, ale nabídneme čtenáři ilustraci o přechodném ději pro různé varianty suboptimálních řešení:



Obr. 39 Výsledky multikriteriální optimalizace pro soustavu o přenosu V_2 (hodnotícími kritérii jsou integrální plocha ITAE pro skok řízení „control ITAE“ a pro skok poruchy „error ITAE“)



Obr. 40 Odezvy regulačního obvodu se soustavou o přenosu F_2 na jednotkový skok řízení a jednotkový skok poruchy pro regulátory: ZN = podle Ziegler-Nicholse, ot.mod = metoda optimálního modulu, a dále pro regulátory nastavené GA 1DOF PID a 2DOF PID (různé hodnoty na Paretově hranici dle obrázku Obr. 39)

6 SYNTÉZA A OPTIMALIZACE LINEÁRNÍCH REGULÁTORŮ

6.1 Klasické a moderní metody syntézy a optimalizace lineárních regulátorů

Skupina lineárních regulátorů je množinou všech regulátorů (systémů), které lze popsat lineárními diferenciálními rovnicemi [4, 11, 13, 20]. Do této skupiny spadá i regulátor PID. Uvážíme-li, že přibližně 90% průmyslových aplikací řídí PID regulátory [27, 28], o zbylých 10% se dělí ostatní lineární regulátory společně s fuzzy regulátory [11], regulátory založené na neuronových sítích, atd. Je tedy patrné, že lineární regulátory jiného typu než PID dosud nenalezly širší uplatnění, ačkoli ze své podstaty nabízejí široké možnosti.

Regulátor PID vyniká kvalitou regulace, jednoduchostí, robustností. S velkou pravděpodobností však nalezneme pro regulovaný systém ve zbytku množiny lineárních regulátorů stejně kvalitní a robustní regulátory, možná dokonce kvalitnější a robustnější. Problém spočívá pouze v komplikovanosti nalezení jejich tvaru, potažmo nastavování jejich parametrů. Pouze některé z běžně používaných metod pro syntézu regulátorů (např. metoda optimálního modulu [4], metoda kvadratické regulační plochy [11], atd.) uvažují obecné lineární struktury regulátoru. Nicméně i u výše uvedených analytických metod syntézy se výpočet stává neúměrně komplikovaný. A to nemluvíme o možnosti multikriteriální optimalizace, kde nastává problém i u optimalizace parametrů regulátoru PID.

Možnosti výpočetní techniky nabízejí přechod od metod analytických k metodám simulačním. Optimální řešení tak není spočteno exaktně, nýbrž hledáno numerickými metodami a meta-heuristikami [9]. Předpokladem úspěchu je znalost chování systému a návazné sestavení simulačního modelu (problematikou identifikace se zabývá například literatura [16]). Metodami získání popisu soustavy se zde nebudeme zabývat, jelikož ten je předpokládán i u většiny klasických metod.

Odhlédneme-li od samotného návrhu struktury a parametrů regulátoru, složitější struktura regulátoru není primárně obtíž v návazném vyšetřování stability a robustnosti regulačního obvodu při působení poruchy, ale působí komplikace v analýze citlivosti na změnu parametrů soustavy [17]. Nicméně i uvedenou analýzu lze pomocí analytických metod (Kharitonovy polynomy) či numerických metod řešit.

6.1.1 Obecný lineární regulátor a jeho optimalizace

Budeme-li uvažovat optimalizaci obecných lineárních regulátorů, stojíme před složitějším problémem, než je například optimalizace parametrů PID. Regulátor PID má totiž pevně danou strukturu a úkolem při jeho optimalizaci je pouze nalezení vhodných parametrů. Při optimalizaci obecného lineárního regulátoru však musíme stanovit samotnou strukturu a následně optimalizovat jeho parametry (návrh struktury a optimalizaci parametrů lze řešit i současně, jak bude ukázáno v popisu autorovy vlastní metody syntézy regulátorů).

Regulátory budeme pro naše účely popisovat pomocí operátorového přenosu, který je poměrem obrazů výstupu a vstupu systému v Laplaceově transformaci [19, 20].

Přenos má obecně tvar poměru dvou polynomů, kdy musí být zachováno pravidlo kauzality, tj. při popisu regulátoru pomocí Laplaceovy transformace bude platit, že stupeň polynomu čitatele přenosu bude menší nebo nanejvýš stejný jako stupeň polynomu ve jmenovateli přenosu (u regulátorů s derivačním charakterem uvažujeme realizační konstanty ve jmenovateli).

$$G_R(s) = \frac{b_m s^m + \dots + b_2 s^2 + b_1 s + b_0}{a_n s^n + \dots + a_2 s^2 + a_1 s + a_0}, m \leq n \quad (13)$$

Cílem je nyní nalézt optimální stupeň polynomů a jejich koeficienty.

6.1.2 Metoda geometrického místa kořene jako inspirace

Hlavní myšlenkou předložené práce je využití jednoduchých metaheuristik pro jednofázovou (simultánní) optimalizaci struktury a parametrů regulátoru (který představuje obecný lineární systém). Necháme-li se inspirovat metodu geometrického místa kořene, můžeme daného cíle dosáhnout.

PŘENOS $F_0(s)$	POČET NUL PÓLŮ	PRŮBĚH GMK	PŘENOS $F_0(s)$	POČET NUL PÓLŮ	PRŮBĚH GMK
$\frac{K_0 s}{T_1 s + 1}$	$\frac{1}{1}$				
$K_0 \frac{T_0 s + 1}{T_1 s + 1}$	$\frac{1}{1}$		$\frac{K_0 (T_0 s + 1)}{(T_1 s + 1)(T_2 s + 1)}$	$\frac{1}{2}$	
$K_0 \frac{T_0 s + 1}{s}$	$\frac{1}{1}$				
$\frac{K_0}{(T_1 s + 1)(T_2 s + 1)}$	$\frac{0}{2}$		$\frac{K_0 (T_0 s + 1)}{T_1^2 s^2 + T_2 s + 1}$	$\frac{1}{2}$	
$\frac{K_0}{(T s + 1)^2}$	$\frac{0}{2}$		$\frac{K_0 s}{T_1^2 s^2 + T_2 s + 1}$	$\frac{1}{2}$	
$\frac{K_0}{T_1^2 s^2 + T_2 s + 1}$	$\frac{0}{2}$		$\frac{K_0}{(T_1 s + 1)(T_2 s + 1)(T_3 s + 1)}$	$\frac{0}{3}$	
$\frac{K_0}{s (T_1 s + 1)}$	$\frac{0}{2}$		$\frac{K_0}{(T s + 1)^3}$	$\frac{0}{3}$	
$\frac{K_0 s}{(T_1 s + 1)(T_2 s + 1)}$	$\frac{1}{2}$		$\frac{K_0}{s (T_1 s + 1)(T_2 s + 1)}$	$\frac{0}{3}$	

Obr. 41 Geometrická místa kořenů charakteristických rovnic v praxi často používaných systémů [53]

Metoda geometrického místa kořene je starší grafickou metodou návrhu regulátorů, kdy do komplexní roviny vyneseme kořeny (póly a nuly) regulované soustavy a dále zde umístíme kořeny regulátoru. Existuje zde řada pravidel, jak se při zvětšujícím zesílení budou “pohybovat” kořeny uzavřeného regulačního obvodu.

Samotná pravidla pro návrh regulátoru pomocí metody geometrického místa kořene nejsou příliš podstatná. Podstatná je myšlenka umístování pólů a nul do komplexní roviny. Jestliže totiž přidáme do komplexní roviny kořen, změníme tak tvar přenosu, pozice kořenu pak ovlivní i výsledné parametry regulátoru.

6.1.3 Efektivita návrhu

Je žádoucí, aby regulátor sám o sobě neobsahoval póly v pravé komplexní polorovině. Neboť jsou-li v otevřeném obvodu kladné póly, stěží dosáhneme stability výsledného regulačního obvodu. Zde se již ukazuje první výhoda metod založených na optimalizaci umístění kořenů, jelikož můžeme zamezit vkládání kladných kořenů do jmenovatele operátorového přenosu.

Pokud bychom totiž optimalizovali koeficienty polynomu, a ne jeho kořeny, museli bychom následně vyšetřit stabilitu přenosu regulátoru. Je vhodné připomenut že kladné koeficienty polynomů jmenovatele jsou pro stabilitu podmínkou nutnou, nikoli však dostačující.

Nutnou a postačující podmínkou pro stabilitu systému je, aby všechny kořeny polynomu ve jmenovateli přenosu (tzv. *póly*) měly zápornou reálnou složku, tj. aby všechny póly systému ležely v levé komplexní polorovině. [4, 11, 20].

6.1.4 Tvary operátorového přenosu

Operátorový přenos je podílem dvou polynomů. Jestliže vyjádříme polynomy ve tvaru kořenových činitelů, můžeme s přenosu číst přímo jednotlivé póly a nuly, mluvíme tak o vyjádření ZPK (zeros-poles-gain, tj. vyjádřením pólů, nul a zesílení).

Dále se nabízí vyjádření ve tvaru časových konstant. Platí, že časové konstanty jsou záporně vzatými převrácenými hodnotami kořenů (neuvažujeme nyní komplexní kořeny, při jejichž výskytu se ve formě časových konstant objeví (v reálném oboru) nerozložitelný polynom druhého stupně). Nicméně pro optimalizační účely nebudeme tvar s časovými konstantami používat. Mezi oběma popisy lze snadno přecházet, ale vyjádření pomocí pólů a nul bude pro naše účely výhodnější.

$$G_R(s) = K \frac{(s - z_1)(s - z_2) \dots (s - z_m)}{(s - p_1)(s - p_2) \dots (s - p_n)} \quad (14)$$

$$G_R(s) = k \frac{(t_1 s + 1)(t_2 s + 1) \dots (t_m s + 1)}{(T_1 s + 1)(T_2 s + 1) \dots (T_n s + 1)} \quad (15)$$

6.2 Vlastní metoda pro syntézu a optimalizaci lineárních regulátorů

6.2.1 Kriteriační funkce

Vzhledem ke skutečnosti, že pro návrh a optimalizaci regulátoru využíváme genetický algoritmus, jakožto jednu z globálních optimalizačních metod, je nutné formulovat úlohu ve formě kriteriační funkce:

$$I = f_k(\mathbf{x}) \quad (16)$$

kde f_k je kriteriační funkce, \mathbf{x} je vektor parametrů regulátoru a I je hodnota kriteriační funkce.

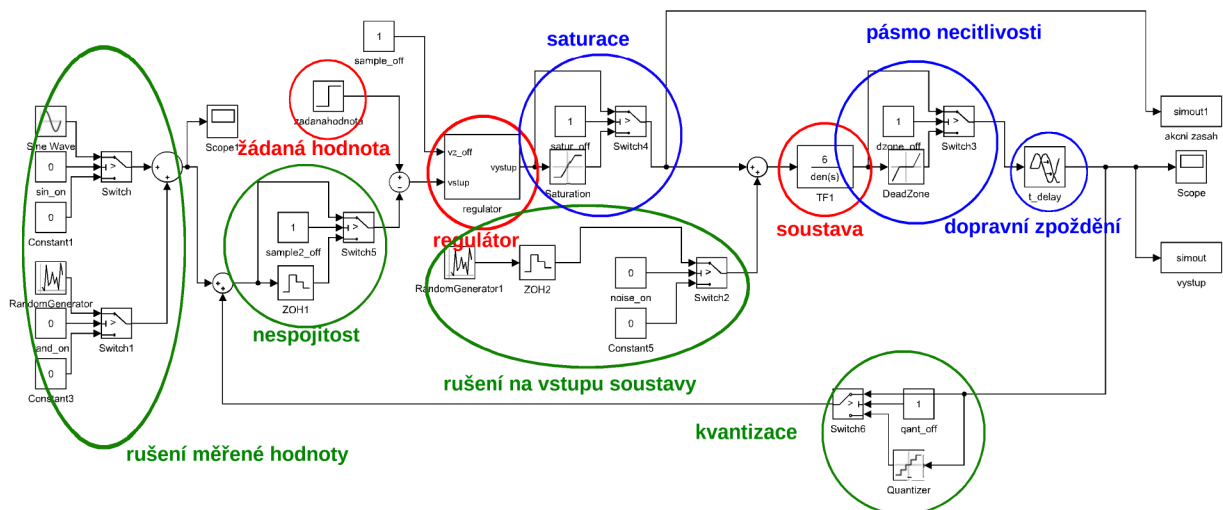
Vektor parametrů regulátoru může nést nejen hodnoty parametrů pevné struktury regulátoru, ale při volbě vhodného kódování může popisovat i samotnou strukturu regulátoru (viz níže).

Stanovení hodnoty kriteriační funkce může být založeno buď na analytickém výpočtu nebo na simulaci. Vzhledem k povaze úlohy je využito výpočtu s pomocí simulace, jelikož kriteriační funkce zahrnuje více dílčích kritérií a rovněž vyčíslení integrální plochy ITAE (což analyticky nelze).

6.2.2 Simulační model

Jádro simulačního modelu tvoří zpětnovazební regulační smyčka obsahující regulátor a soustavu. Vstupem regulační smyčky je žádaná hodnota ve tvaru jednotkového skoku, regulovaná veličina je pak vyvedena na zobrazovací prvek. Průběh všech veličin je současně exportován do prostředí Matlab pomocí bloku *To Workspace* (viz blok *simout*).

Volitelně mohou aktivovány další části regulačního obvodu: Na výstupu regulátoru může být aktivováno omezení akčního zásahu. Na výstupu soustavy lze aktivovat dopravní zpoždění. Rovněž lze aktivovat rušení na vstupu do regulované soustavy, či rušení ve zpětné vazbě (např šum na snímači). Posledním doplňkem je volitelné aktivování kvantifikace hodnoty (z čidla) ze zpětné vazbě.



Obr. 42 Nastavitelné simulační schéma pro účely optimalizace lineárních regulátorů

6.2.3 Průběh optimalizačního procesu

Optimalizační proces se skládá z inicializační, optimalizační a vizualizační části. Inicializační část zahrnuje nastavení parametrů simulace, jejich ověření a aplikování do simulačního modelu. Dále s v inicializační části provádí nastavení parametrů řešiče, tj. genetického algoritmu. Následuje běh optimalizační smyčky, po jehož skončení jsou dosažené výsledky vypsány a vizualizovány.

Následující ilustrace vyobrazuje základní vývojový diagram vytvořeného software pro optimalizaci obecných lineárních regulátoru (software je přiložen k práci).

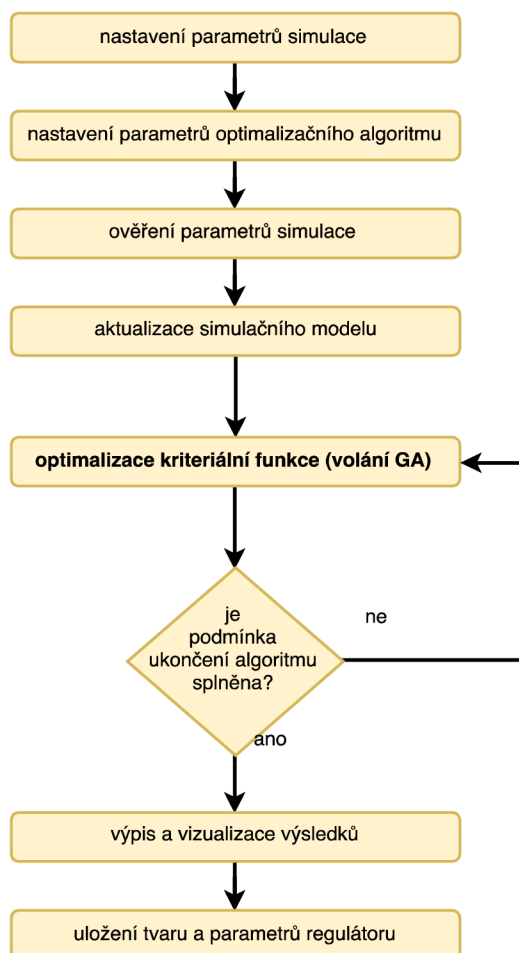
Algoritmus zobrazený ve vývojovém diagramu níže je implementován v zdrojovém souboru *optimalizace.m*. Algoritmus je možné volat jako funkci ve tvaru:

$[GAR, reg, Z, P, K, KID, Ts, FVAL]=optimalizace(system_tf, count_iter, t_sim, gar_type)$

Vstupními parametry funkce jsou: přenos soustavy *system_tf*, maximální dovolený počet iterací (generací GA) *count_iter*, simulační čas *t_sim* a typ regulátoru *gar_type*. Podporované typy regulátorů:

- *PID*
- *GAR3K* – obecný lineární regulátor (nejvýše 3 stupně)
- *GAR5R* – lineární regulátor s čistě reálnými kořeny (nejvýše 5 stupně)
- *GAR5K* – obecný lineární regulátor (nejvýše 5 stupně)

Výstupními hodnotami funkce jsou: datová struktura obsahující všechny nastavení a výsledky (pro možnost uložení) *GAR*, výsledný tvar nalezeného regulátoru ve formě operátorového přenosu *reg*, Dále vektor nul *Z* a vektor pólů *P* regulátoru, zesílení regulátoru *K*, parametry PID (pokud byl vybrán odpovídající typ regulátoru) *KID*, vzorkování simulace *Ts* a nejlepší dosažená hodnota kritériální funkce *FVAL*.



Obr. 43 Základní vývojový diagram softwaru pro optimalizaci lineárního regulátoru

Nastavení parametrů simulace zajišťují funkce *gar_basic_settings* a *gar_additional_settings*. Funkce *gar_update_solver* zajistí nastavení parametrů řešiče GA. Funkce *gar_test_model_settings* ověří platnost nastavených parametrů simulace. Pro aktualizaci simulačních modelů slouží funkce *gar_update_model* resp. *gar_update_modelPID* (v případě oprimalizace PID). Funkce *gar_results_printing* slouží pro výpis a vizualizaci výsledků. Funkce *gar_reg_rek* pak slouží pro rekonstrukci tvaru regulátoru z předloženého genotypu genetického řetězce.

6.2.4 Forma a princip kódování lineárních regulátorů

Je zapotřebí polohy kořenů systému zakódovat vhodným způsobem do genetického řetězce. Popíšeme nejprve jednodušší verzi zahrnující pouze regulátory s čistě reálnými kořeny a následně přístup rozšíříme na všechny lineární regulátory. Cílem je nalézt vhodné kódování, pomocí něhož bychom dosáhli současně optimalizace struktury i parametrů regulátoru.

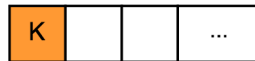
Kódování parametrů lineárního regulátoru v rámci jedince GA (řetězce GA) je dáno souborem pravidel, která byla pro tento účel vytvořena a jsou základem know-how představeného přístupu. Přístup vychází z myšlenky geometrického místa kořenů operátorového přenosu, kdy pro stabilní přenos musí platit, že všechny póly (kořeny jmenovatele přenosu) mají zápornou reálnou část, tj. leží v levé části komplexní roviny. Nebylo by problematické optimalizovat kořeny pouze v rozsahu záporných hodnot, ale tímto způsobem bychom nedosáhli kýženého cíle modifikace struktury regulátoru.

6.2.4.1 Verze regulátoru s (výhradně) reálnými kořeny.

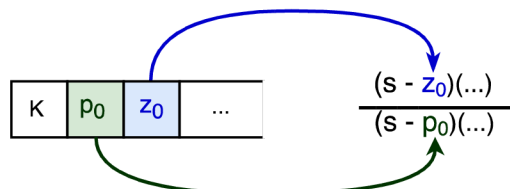
V publikaci [1] tuto variantu pracovně označujeme GAR n , kde n představuje maximální stupeň polynomu jmenovatele. Stupeň polynomu čitatele je rovný nebo nižší než stupeň polynomu jmenovatele. Pokud během syntézy nastane situace, že polynom čitatele je větší než polynom jmenovatele, pak je polynom ve jmenovateli rozšířen o další kořeny (póly) s vysokou dynamikou (kořeny s velmi vysokou dynamikou mají na regulační děj malý vliv). Platí, že každý kořen je kódován právě jedním genem, mimoto je jeden gen vyhrazen pro kódování zesílení regulátoru.

Pro účely optimalizace tvaru regulátoru byla použita celá komplexní rovina s tím, že kořeny rozmísťujeme v rámci celé roviny, nicméně pro syntézu se uplatní pouze ty kořeny, které mají zápornou reálnou část, tj. v případě GAR n jsou záporné. Shrňme tedy pravidla návrhu:

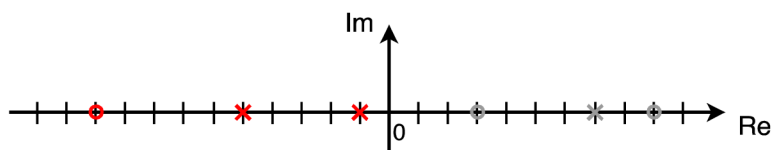
- a) Na začátku genetického řetězce se nalézá gen kódující zesílení přenosu. Hodnota zesílení je optimalizována pouze v kladných hodnotách.



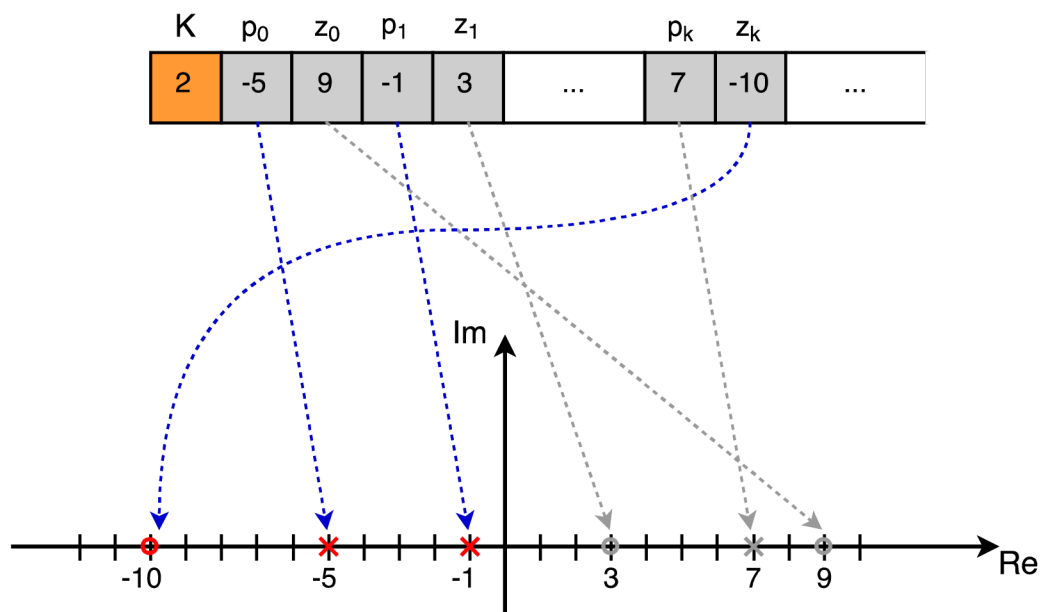
- b) Od druhé pozice jsou v genetickém řetězci kódovány jednotlivé kořeny. Je potřeba odlišit nuly a póly (tj. kořeny náležící čitateli a kořeny náležící jmenovateli). V našem případě bylo zvoleno alternující uspořádání, kdy se v řetězci střídá vždy jeden pól a jedna nula. Toto uspořádání je výhodnější z pohledu operace křížení (prováděného genetickým algoritmem), na pořadí kořenů při procesu syntézy jinak vůbec nezáleží.



- c) Kořeny ležící v levé polorovině jsou použity pro syntézu lineárního regulátoru, zatímco kořeny, které leží v pravé části komplexní poloroviny, se v procesu syntézy neuplatní.

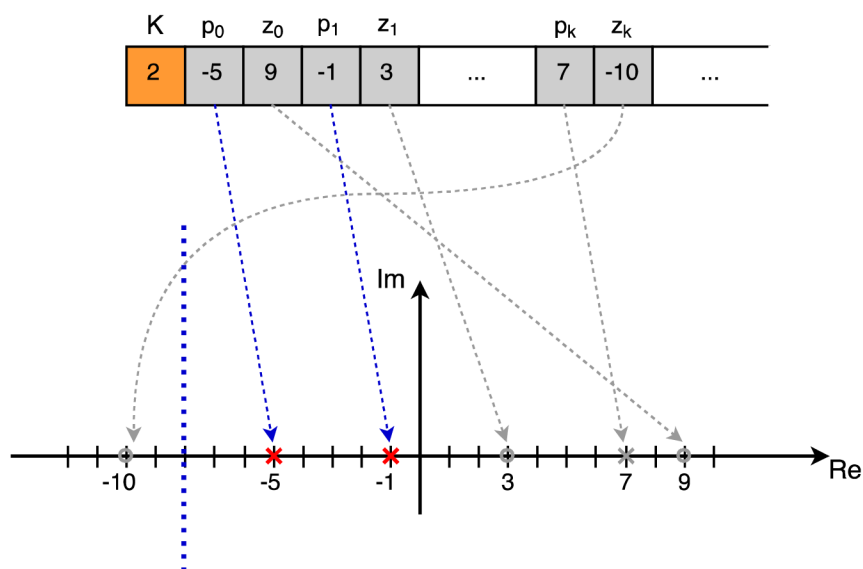


Pro kódování regulátoru GARn platí, že kódující řetězec má délku $2n+1$. Následuje ilustrace interpretující proces kódování regulátorů GARn.



Obr. 44 Grafická interpretace kódování/dekódování regulátoru typu GAR do/z řetězce GA. Pro eliminaci kořenů byla využita pravá komplexní polorovina. Kořeny, které spadly do kladné části komplexní roviny, se při syntéze regulátoru nevyužijí a tím je ovlivněn i výsledný tvar regulátoru.

Metoda použití pravé části komplexní poloroviny za účelem eliminace kořenů je první testovanou variantou. Druhou a z mnoha pohledů výhodnější variantou by bylo určení maximální dynamiky regulátoru, kterou má smysl uvažovat. Kořeny jsou pak umisťovány pouze do levé komplexní poloroviny, ale kořeny které leží za hranicí (maximální uvažované dynamiky) jsou eliminovány. Případně je možné oba přístupy kombinovat.



Obr. 44 Grafická interpretace kódování/dekódování regulátoru typu GAR do/z řetězce GA. Pro eliminaci kořenů byla využita pravá komplexní polorovina a hranice stanovené maximální uvažované dynamiky. Kořeny, které spadly do kladné části komplexní roviny nebo za výše zmíněnou hranici, se při syntéze regulátoru nevyužijí a tím je ovlivněn i výsledný tvar regulátoru.

V některých případech je žádoucí, aby regulátor obsahoval integraci, tj. nulový kořen. Integrační složka sice zhoršuje stabilitu uzavřené smyčky, ale díky integračnímu charakteru, který je do obvodu zaveden, jsme schopni dosáhnout nulové ustálené regulační odchylky i v případě působení poruchy. Integrační složku je možné při optimalizaci nechat vzniknout jako produkt optimalizace (s tím, že velmi malé hodnoty kódující kořen jsou interpretovány jako nulové), v případě že víme, že je vhodné integraci přidat, můžeme tak učinit i manuálně (integrační složka bude přidána automaticky při syntéze regulátoru). Přenos pak bude mít následující tvar:

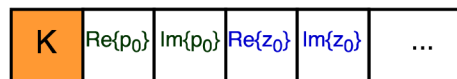
$$G_{\bar{R}}(s) = K \frac{(s-z_1)(s-z_2)\dots(s-z_m)}{s(s-p_1)(s-p_2)\dots(s-p_n)}, m \leq n+1 \quad (17)$$

6.2.4.2 Obecná verze regulátoru s komplexními kořeny

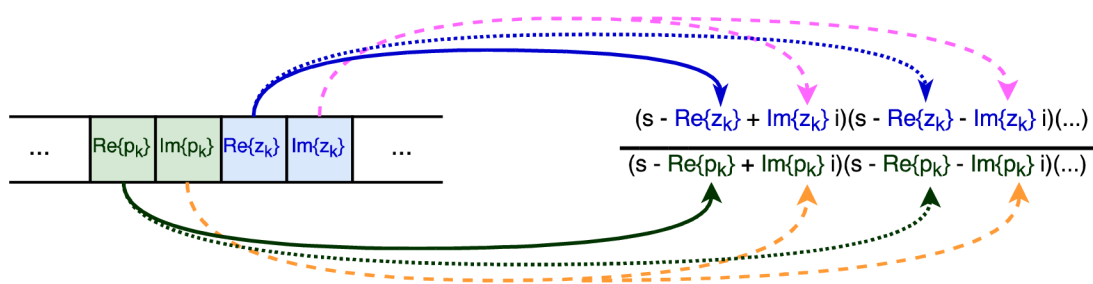
1. Obecnou verzi lineárního regulátoru s komplexními kořeny jsme v publikaci [1] pracovně označili GARnC. Komplexní kořen již není možno kódovat pouze jedním genem, ale pro kódování kořenu jsou použity vždy dva geny, přičemž první gen kóduje reálnou část kořene a druhý gen kóduje imaginární část kořene. V kódování je třeba navíc zajistit, aby se komplexní kořeny vyskytovaly pouze jako komplexně sdružené dvojice. Kromě kořenů je v genetickém řetězci (na jeho začátku) kódováno výsledné zesílení regulátoru.

Pro účely kódování regulátoru GARnC byl vytvořen soubor pravidel, který je rozšířením metody návrhu regulátorů GARn. Opět je využito celé komplexní roviny, jako základu know-how procesu syntézy:

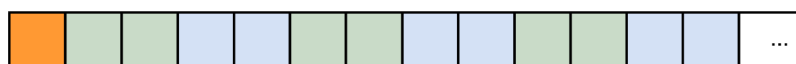
1. Na začátku genetického řetězce leží gen kódující zesílení. Zesílení je optimalizováno pouze v kladných hodnotách.



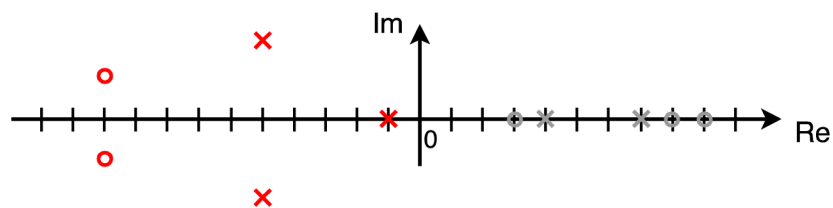
2. Následují dvojice genů kódujících jednotlivé kořeny. První z dvojice genů kóduje reálnou složku, druhý pak imaginární složku kořene.



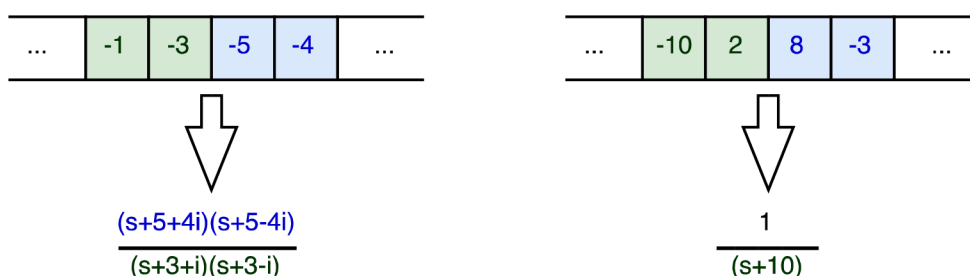
3. V řetězci jsou póly a nuly uspořádány v alternujícím pořadí, tj. vždy jeden pól a jedna nula, následuje další pól a další nula, atd.



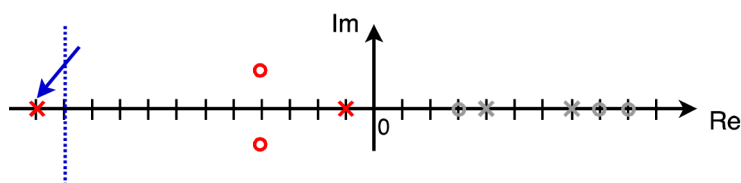
4. Je-li reálná část kořene kladná, kořen se v procesu syntézy neuplatní (imaginární část v tomto procesu eliminace nehraje žádnou roli). Pokud je reálná část nekladná, postupujeme dle následujícího pravidla.



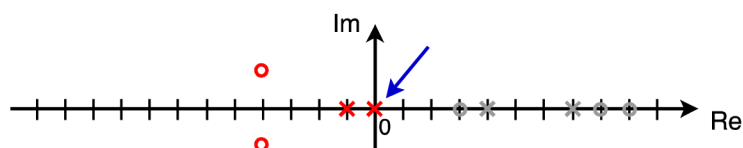
5. Jestliže je hodnota imaginární složky kladná (či nulová), imaginární část kořene bude ignorována a bude přidán pouze jeden reálný kořen, určený svou reálnou částí. Pokud je hodnota genu kódujícího imaginární část záporná, budou vygenerovány dva komplexně sdružené póly, přičemž velikost imaginární složky bude dána absolutní hodnotou kódujícího genu (reálná složka obou kořenů pak bude určena genem kódujícím reálnou část kořene).



6. V případě, že by počet nul v procesu syntézy překročil počet pólů, doplníme přenos o póly s vysokou dynamikou, aby byl výsledný přenos kauzální.



7. Je-li žádoucí přidat do regulátoru integrační člen, můžeme tak učinit obdobně jako u GARn.



Počet genů v řetězci kódujícím regulátor GARnC je $4n+1$.

I v případě GARnC je možné též uvažovat modifikaci přístupu a pro umístování kořenů uvažovat pouze levou část komplexní poloroviny, s tím že bude definována hranice dynamiky, a kořeny s vyšší dynamikou se neuplatní (pracovně označme tyto kořeny jako ležící v *šedé zóně*).

Využití šedé zóny by oproti stávajícímu přístupu nabídlo možnost povolit i generování přenosů s neminimální fází. Výše popsané kódování GARn ani GARnC vznik regulátorů s neminimální fází neumožňuje.

6.2.5 Úprava zesílení lineárního regulátoru

Princip kódování nastíněný v kapitole 3.2.3 nabízí efektivní způsob kódování, který umožňuje i modifikaci struktury regulátoru. Přístup je použitelný sám o sobě, avšak drobnými úpravami v kódování jsme schopni docílit výrazného zvýšení efektivity optimalizace. Jako významný krok ve zvyšování efektivity je využití úpravy zesílení regulátoru GARn resp. GARnC. Uvedený princip je podstatnou součástí know-how metody syntézy.

Uvažujme nyní přenos ve formě ZPK. Přenos je tvořen jednotlivými kořenovými činiteli. Jestliže mezi sebou jednotlivé kořenové činitele roznásobíme, obdržíme výsledný tvar přenosu ve formě podílu dvou polynomů. Výsledné zesílení je dáno poměrem absolutních členů polynomu čitatele a polynomu jmenovatele. Toto výsledné zesílení je určeno jednak zesílením K , které kódujeme v genetickém řetězci, a rovněž zesílením K_{zp} , které je podílem produktu nul a produktu pólů.

$$k = \frac{b_0}{a_0} = K K_{zp} = K \frac{\prod_{j=1}^m z_j}{\prod_{i=1}^n p_i} \quad (18)$$

Z uvedeného však plynou jisté komplikace:

1. Výsledné zesílení není možné v uvedené formě optimalizovat jako samostatný parametr, neboť je ovlivněno zesílením, které do přenosu vnáší jednotliví kořenoví činitelé.
2. Jestliže dojde k vzniku nebo zániku některého kořene dojde v zesílení k výraznému skoku (pracovně mluvíme o „nárazu“ ve změně zesílení).
3. Kořeny, které se nacházejí daleko od počátku souřadného systému, představují kořeny s vysokou dynamikou (a jsou pro regulační děj méně významné). Z toho vyplývá, že při daném kódování málo dominantní kořeny významně ovlivňují výsledné zesílení.

Pro prakticky každý optimalizační algoritmus je výhodnější, pokud je účelová funkce „ukázněná“, tj. při malé změně vstupních proměnných se nemění výstupní hodnoty dramatickým způsobem. Je proto žádoucí, abychom v maximální míře eliminovali veškeré „nárazy“ a závislosti na málo významných parametrech. Současně by bylo vhodné, abychom mohli zesílení regulátoru nastavovat jako nezávislý parametr.

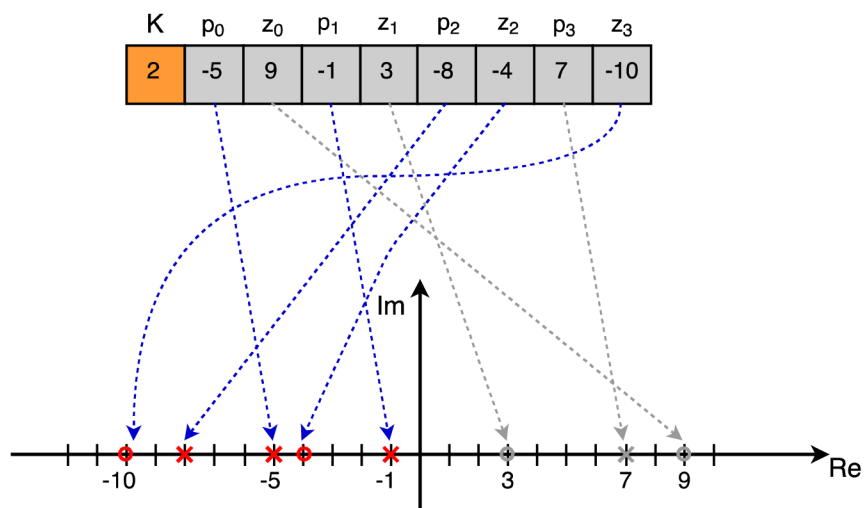
Naším cílem je proto eliminovat vliv polohy kořenů na výsledné zesílení regulátoru. Vyjdeme-li ze vztahu (11), požadujeme, aby zesílení bylo určeno pouze hodnotou K , nikoli hodnotou K_{zp} .

Chceme-li se na provedenou operaci podívat blíže, vyjdeme z nosné myšlenky. Provedme jednoduchou operaci tak, že každý kořenový činitel podělíme absolutní hodnotou jeho kořene a následně všechny takto vytknuté konstanty sloučíme do jediného koeficientu. V podstatě docílíme skutečnosti, že převedeme přenos do tvaru, který připomíná tvar zápisu s časovými konstantami.

$$\begin{aligned} G_R(s) &= K \frac{(s-z_1)(s-z_2)\dots(s-z_m)}{(s-p_1)(s-p_2)\dots(s-p_n)} = K \frac{\prod_{j=1}^m -z_j \left(\frac{1}{-z_1}s+1\right)\left(\frac{1}{-z_2}s+1\right)\dots\left(\frac{1}{-z_m}s+1\right)}{\prod_{i=1}^n -p_i \left(\frac{1}{-p_1}s+1\right)\left(\frac{1}{-p_2}s+1\right)\dots\left(\frac{1}{-p_n}s+1\right)} \quad (19) \\ &\dots = K K_{zp} \frac{(t_1s+1)(t_2s+1)\dots(t_ms+1)}{(T_1s+1)(T_2s+1)\dots(T_ns+1)} = k \frac{(t_1s+1)(t_2s+1)\dots(t_ms+1)}{(T_1s+1)(T_2s+1)\dots(T_ns+1)} \end{aligned}$$

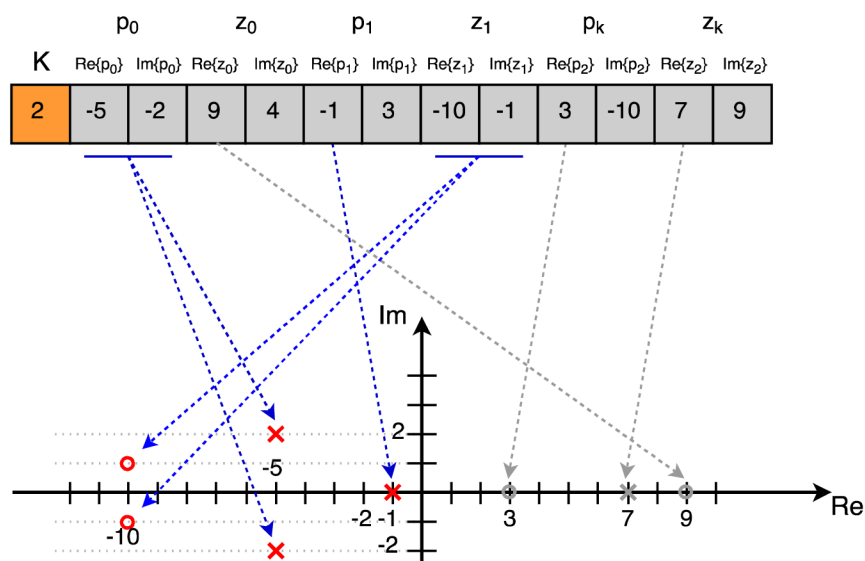
Vhodným krokem je tedy podělit daný vztah právě hodnotou K_{zp} . Praktický dopad bude, že polohou kořenů ovlivňujeme pouze dynamiku, nikoli zesílení, které je optimalizováno samostatně.

$$G_{R\,final}(s) = K \frac{\prod_{i=1}^n \|-p_i\| \frac{(s-z_1)(s-z_2)\dots(s-z_m)}{(s-p_1)(s-p_2)\dots(s-p_n)}}{\prod_{j=1}^m \|-z_j\|} \quad (20)$$



$$G_R(s) = K \frac{\prod_{i=1}^n \|-p_i\| (s-z_1)(s-z_2)\dots(s-z_m)}{\prod_{j=1}^m \|-z_j\| (s-p_1)(s-p_2)\dots(s-p_n)} = 2 \cdot \frac{5 \cdot 1 \cdot 8}{4 \cdot 10} \frac{(s+4)(s+10)}{(s+5)(s+1)(s+8)}$$

$$= 2 \frac{(s+4)(s+10)}{(s+5)(s+1)(s+8)} = \frac{2s^2 + 28s + 80}{s^3 + 14s^2 + 53s + 40}$$



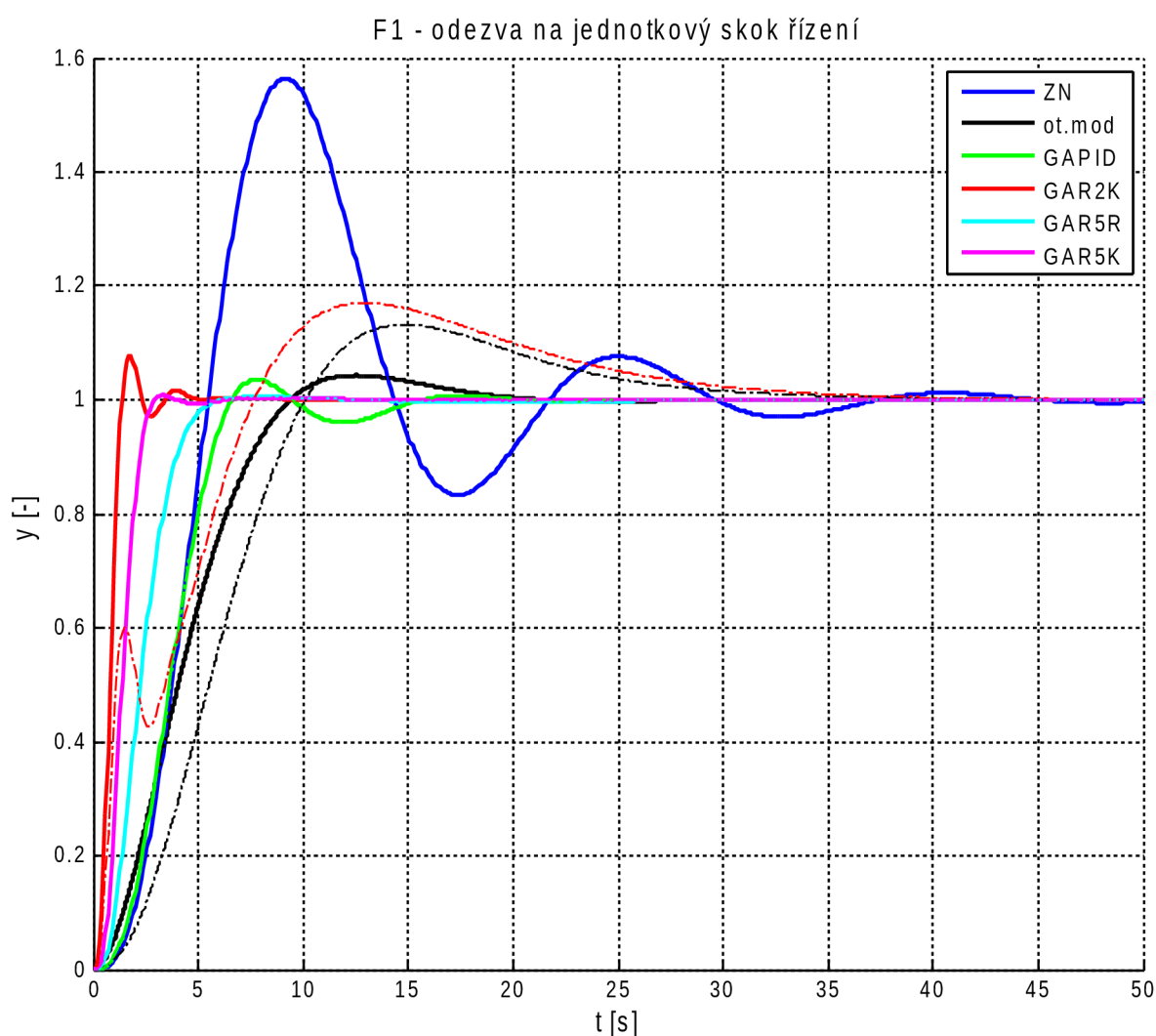
$$G_R(s) = K \frac{\prod_{i=1}^n \|-p_i\| (s-z_1)(s-z_2)\dots(s-z_m)}{\prod_{j=1}^m \|-z_j\| (s-p_1)(s-p_2)\dots(s-p_n)} = 2 \cdot \frac{2 \cdot \sqrt{5^2 + 2^2} \cdot 1}{2 \cdot \sqrt{10^2 + 1^2}} \frac{(s+10+i)(s+10-i)}{(s+5+2i)(s+5-2i)(s+1)}$$

$$= \frac{2s^2 + 40s + 202}{s^3 + 11s^2 + 39s + 29}$$

6.2.6 Výsledky syntézy GAR pro vybrané soustavy

Základní myšlenky vlastní metody (označené GAR) pro syntézu obecných lineárních regulátorů byly poprvé prezentovány v roce 2011 ve Francii na mezinárodní konferenci *World Academy of Science Engineering and Technology*. Za dobu vývoje prošla vytvořená softwarová knihovna řadou modifikací optimalizujícími efektivitu, částečně optimalizováno bylo i kódování a rozšířeny byly též možnosti nastavení kritériální funkce.

Pro prezentaci dosažených výsledků uvedme výsledky optimalizace regulátorů pro regulační obvod se soustavami o přenosu F_1 a V_1 , v případě prvně uvedené soustavy je možno srovnat nalezená řešení s metodou podle Ziegler-Nicholse a metodou optimálního modulu. V případě soustavy V_1 stojíme před problematikou regulace soustavy s velkým dopravním zpožděním (dopravním zpožděním je zde větší než dominantní časová konstanta), což činí úlohu komplikovanější, tudíž výsledky jsou o to více zajímavé. Popis a vlastnosti uvedených testovacích soustav jsou k dispozici v kapitole 5.



Obr. 46 Srovnání regulátorů na základě přechodových charakteristik zpětnovazebního zapojení soustavy F_1 s regulátorem; ZN=PID nastavené metodou Ziegler-Nichols, ot.mod=PID nastavené podle metody optimálního modulu, GAPID=PID seřizené na nejmenší plochu ITAE pomocí GA, GAR2K=lineární regulátor s maximální délkou polynomu 3 (i komplexní kořeny), GAR5R= lineární regulátor s maximální délkou polynomu 5 (reálné kořeny), GAR5K= lineární regulátor s maximální délkou polynomu 5 (i komplexní kořeny)

$$G_{ZN}(s) = \frac{1,58s^2 + 1,00s + 0,16}{0,01s^2 + s} \quad (21)$$

$$G_{optmod}(s) = \frac{s^2 + 0,42s + 0,04}{0,01s^2 + 1s} \quad (22)$$

$$G_{GAPID}(s) = \frac{2,32s^2 + 0,65s + 0,064}{0,1s^2 + s} \quad (23)$$

$$G_{GAR2K}(s) = \frac{122,90s^3 + 116,70s^2 + 32,27s + 2,79}{s^3 + 3,92s^2 + 12,37s} \quad (24)$$

$$G_{GAR5R}(s) = \frac{76,35s^4 + 132,70s^3 + 78,24s^2 + 18,25s + 1,36}{s^5 + 8,85s^4 + 28,63s^3 + 39,79s^2 + 19,70s} \quad (25)$$

$$G_{GAR5K}(s) = \frac{219,90s^4 + 906,00s^3 + 661,20s^2 + 173,40s + 13,86}{s^5 + 10,09s^4 + 46,64s^3 + 114,70s^2 + 118,70s} \quad (26)$$

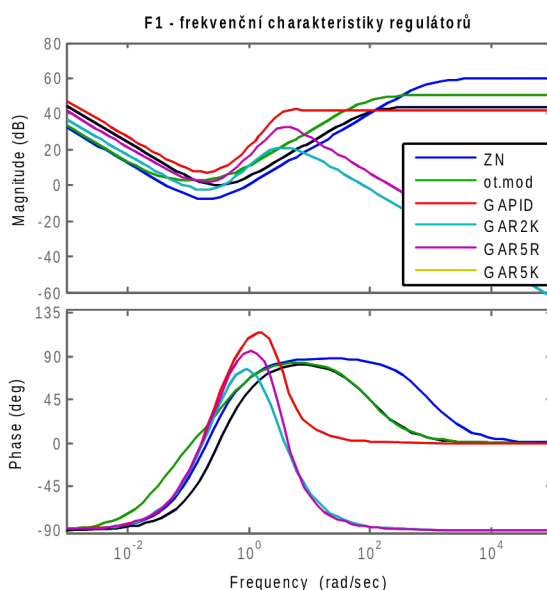
Na obrázku 46 jsou uvedeny přechodové charakteristiky pro regulační obvod se soustavou o přenosu F_1 a jednotlivými typy regulátorů. Z grafu je patrné, že PID nastavené pomocí GA dosahuje v rámci přechodových charakteristik (za daných hodnotících kritérií, tj. ITAE) lepších výsledků než například PID regulátor nastavený podle Ziegler-Nicholse nebo regulátor navržený metodou optimálního modulu. Mnohem lepších výsledků než PID pak dosahují obecné lineární regulátory. Za povšimnutí stojí i skutečnost, že regulátor navržený metodou optimálního modulu a lineární regulátor GAR2K předpokládají výrazně větší akční zásahy než ostatní regulátory, pro zajímavost byla do obvodu přidána saturace o velikosti ± 50 , u obvodů s uvedenými dvěma regulátory se zhoršila dynamika (viz přechodové charakteristiky vynesené čerchovanou čarou), kdežto u ostatních regulátorů zůstaly přechodové charakteristiky nezměněné.

Skutečnost omezených akčních zásahů je dennodenní realitou, neboť saturace představuje společně se třením a pásmem necitlivosti nejčastější nelinearity. Uvažování nelinearit v obvodu se již do jisté míry vymyká této práci, přesto bylo jejich zahrnutí v rámci výzkumu a testování uvažováno.

Vlastnosti přechodové charakteristiky Regulace soustavy F_1	Ziegler Nichols	Optimáln í modul	GA PID	GAR2K	GAR5R	GAR5K
Maximální hodnota [-]	1,5640	1,0432	1,0355	1,0766	1,0053	1,0081
Překmit [%]	56,40%	4,32%	3,55%	7,66	0,53%	0,81%
Doba náběhu (0 – 90%) [s]	5,2	7,5	4,6	1,3	4,0	2,3
Doba ustálení (tolerance 2%) [s]	34,8	16,9	13,4	2,9	4,9	2,5
Integrální krit. ITAE [-]	68,31	15,84	10,16	0,88	4,77	1,84
Integrální krit. ISE (kvadratické) [-]	4,37	3,00	1,96	0,58	1,70	1,05
Max. akční zásah [-]	1,1	1000,4	23,1	122,9	7,6	22,7

Tabulka 4: Vybrané parametry regulačních obvodů pro soustavu F_1 s navrženými regulátory

V následujícím grafu ještě uvedeme frekvenční charakteristiky jednotlivých regulátorů. U regulátorů s derivačním charakterem je uvažována malá realizační (filtrační) konstanta. U regulátoru PID laděného pomocí GA byla použita výrazně větší filtrační konstanta (pro dané nastavení bylo dosaženo lepších výsledků).



Obr. 47 frekvenční charakteristiky regulátorů pro soustavu F_1

6.2.7 Výsledky syntézy GAR pro soustavu s dopravním zpožděním

Druhou ukázkou použití lineárních regulátorů (syntetizovaných genetickými algoritmy) je regulace soustavy o přenosu V_1 , tj. soustavy s velkým dopravním zpožděním. Zde obecné lineární regulátory též dosahují lepších výsledků než regulátory klasické, rozdíl oproti dobře nastavenému PID však již není tak markantní. Výsledky optimalizace jsou prezentovány ve formě přechodových charakteristik, akčních zásahů, tabulky vlastností přechodových charakteristik a v neposlední řadě jsou zde uvedeny frekvenční charakteristiky jednotlivých regulátorů. Přenosy regulátorů jsou následující:

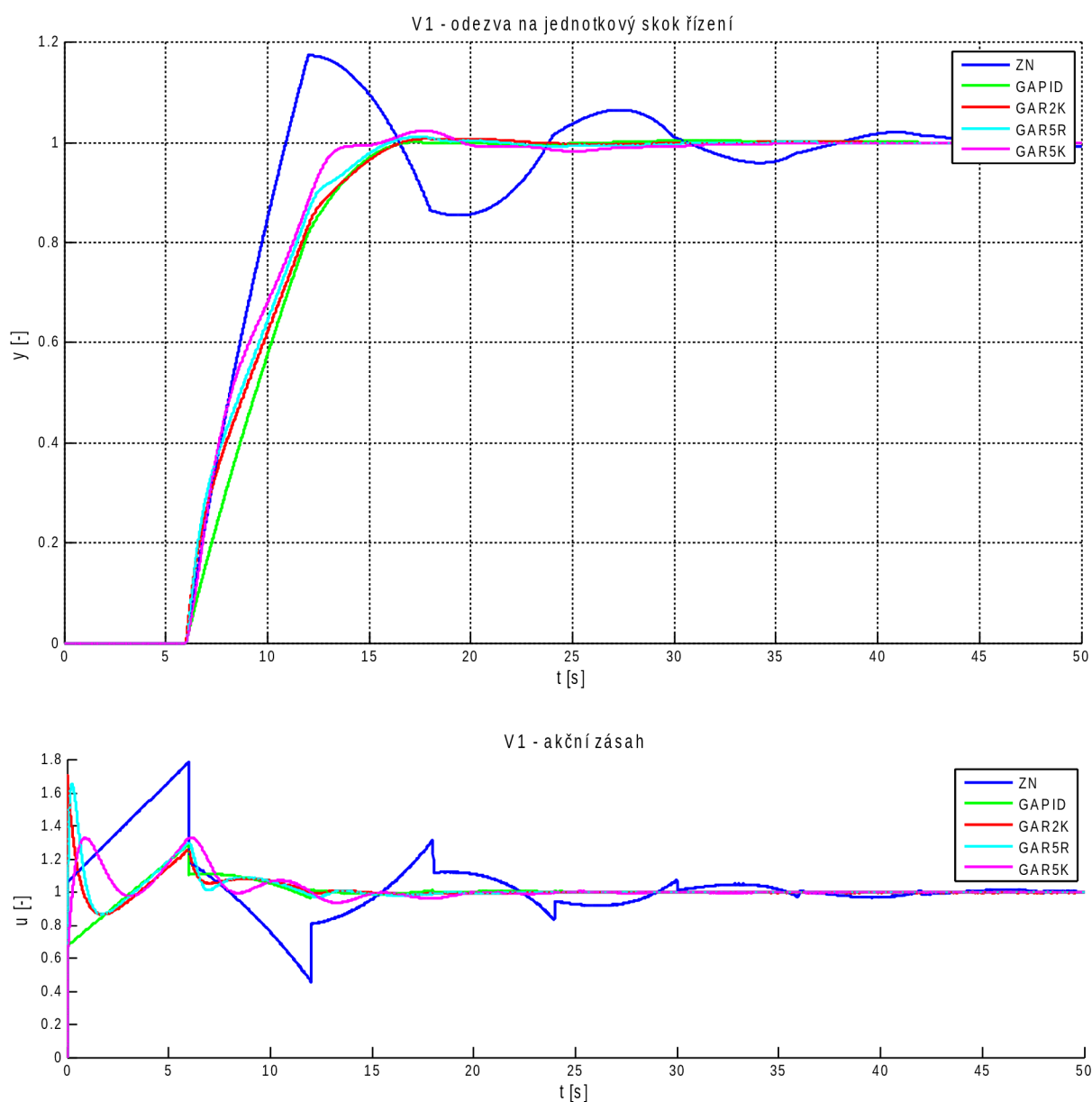
$$G_{ZN}(s) = \frac{2,30 p^2 + 1,06 p + 0,12}{0,01 s^2 + s} \quad (27)$$

$$G_{GAPID}(s) = \frac{1,09 s^2 + 0,67 s + 0,10}{0,01 s^2 + s} \quad (28)$$

$$G_{GAR2K}(s) = \frac{1,72 s^2 + 1,09 s + 0,17}{s^2 + 1,61 s} \quad (29)$$

$$G_{GAR5R}(s) = \frac{39,23 s^4 + 538,8 s^3 + 1457 s^2 + 770,4 s + 113,3}{s^5 + 41,27 s^4 + 338,1 s^3 + 1015 s^2 + 1034 s} \quad (30)$$

$$G_{GAR5K}(s) = \frac{0,54 s^6 + 5,25 s^5 + 14 s^4 + 17,01 s^3 + 10,51 s^2 + 3,14 s + 0,34}{s^6 + 5,34 s^5 + 12,14 s^4 + 14,47 s^3 + 10,01 s^2 + 3,02 s} \quad (31)$$

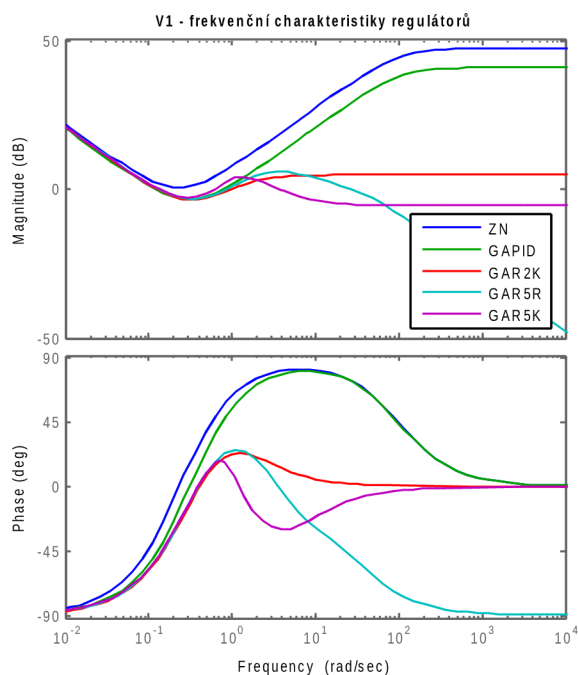


Obr. 48 Srovnání regulátorů na základě přechodových charakteristik zpětnovazebního zapojení soustavy V_1 s regulátory a odpovídající akční zásahy; ZN=PID nastavené metodou Ziegler-Nichols, ot.mod=PID nastavené podle metody optimálního modulu, GAPID=PID seřizené na nejnižší plochu ITAE pomocí GA, GAR2K=lineární regulátor s maximální délkou polynomu 3 (i komplexní kořeny), GAR5R= lineární regulátor s maximální délkou polynomu 5 (reálné kořeny), GAR5K= lineární regulátor s maximální délkou polynomu 5 (i komplexní kořeny)

Vlastnosti přechodové charakteristiky Regulace soustavy V_1	Ziegler Nichols	GA PID	GAR2K	GAR5R	GAR5K
Maximální hodnota [-]	1,1750	1,0036	1,0082	1,0116	1,0238
Překmit [%]	17,50%	0,36%	0,82%	1,16%	2,38%
Doba náběhu (0 – 90%) [s]	10,4	13,4	13,2	12,5	12,2
Doba ustálení (tolerance 2%) [s]	41,4	15,3	15,4	15,0	18,2
Integrální krit. ITAE [-]	78,20	52,47	48,86	47,12	46,75
Integrální krit. ISE (kvadratické) [-]	7,59	8,24	7,78	7,66	7,60
Max. akční zásah (absolutní) [-]	1,8	1,3	1,7	1,7	1,3

Tabulka 5: Vybrané parametry regulačního obvodu se soustavou V_1 a navrženými regulátory

Na obrázku 49 jsou uvedeny frekvenční charakteristiky jednotlivých regulátorů. U regulátorů s derivační složkou je uvažována malá realizační (filtrační) konstanta.



Obr. 49 frekvenční charakteristiky regulátorů pro soustavu V_1

7 ZÁVĚR

Hlavním přínosem práce je vytvoření nové metody pro generování a optimalizaci obecného lineárního regulátoru. Návrhem vhodného způsobu kódování informace bylo docíleno skutečnosti, že tvar i parametry regulátoru jsou společně kódovány v jednom genetickém řetězci a jsou optimalizovány současně. Metoda se tedy liší od běžného postupu, kdy je pomocí gramatické evoluce vygenerována struktura regulátoru a následně jsou pomocí genetického algoritmu optimalizovány parametry již fixní struktury. Navržený přístup umožňuje optimalizovat strukturu regulátoru společně s parametry, což je velkou výhodou ve dvou ohledech. Za prvé, struktura regulátoru se může během optimalizačního procesu průběžně měnit. Za druhé, k optimalizaci lze využít i jednoduché metaheuristiky (genetický algoritmus, simulované žíhání, HC12, atd.) a není třeba použít gramatickou evoluci či databázi předdefinovaných tvarů regulátorů. V druhé zmíněné přednosti spatřuje autor největší sílu vytvořené metody.

Metoda byla inspirována návrhem regulátoru v komplexní rovině s využitím metody geometrického místa kořenů *root locus*. Navržená metoda však z originálního přístupu *root locus* nevyužívá ani grafický přístup ani pravidla pro určení polohy kořenů charakteristické rovnice uzavřeného obvodu. Inspirace je pouze pro kódování: místa kořenů (pólů a nul) regulátoru jasně určují tvar regulátoru. Jestliže pak metoda umožní vznikání a zanikání kořenů a jejich umístování do komplexní roviny, je generován současně tvar i parametry regulátoru.

Metoda má praktický přínos, neboť lze poměrně efektivním způsobem generovat regulátory zajišťující vysokou kvalitu regulace, a to i s využitím jednoduchých optimalizačních řešičů.

Dále má metoda přínos pedagogický, neboť je prezentováno, jak vysoký význam má samotné kódování informace. Volbou vhodného kódování lze docílit vyšší efektivity a nižší výpočetní náročnosti celého optimalizačního procesu.

Metoda byla testována na mnoha různých tvarech a typech regulovaných soustav, samotná práce prezentuje výsledky na několika vhodně zvolených soustavách. Kvalita regulace je klasifikována primárně na základě integrálního kritéria ITAE, a to jak pro odezvu na skok řídicí veličiny, tak na skok poruchy.

Práce rovněž diskutuje problematiku multikriteriální optimalizace. Srovnává regulátor PID s jedním a se dvěma stupni volnosti. Jelikož nejsou stanoveny omezující podmínky, výsledky jsou zobrazeny ve tvaru Paretových hranic, tj. křivek na kterých leží vzájemně nedominantní optimální řešení. Na většině testovaných soustav vykazoval PID regulátor se dvěma stupni volnosti lepší výsledky než PID regulátor s jedním stupněm volnosti, což bylo očekáváno, jelikož PID se dvěma stupni volnosti bude v nejhorším případě vykazovat stejné výsledky jako PID s jedním stupněm volnosti.

8 SEZNAM PUBLIKACÍ AUTORA

MATOUŠEK, R., LANG, S. *Optimalizace parametrů PID regulátoru pomocí GA toolboxu*. In Bratislava: RT Systems, 2010. s. 69-73. ISBN: 978-80-970519-0-7.

MATOUŠEK, R., MINÁŘ, P., LANG, S., PIVOŇKA, P. HC12: *Efficient Method in Optimal PID Tuning*. Lecture Notes in Engineering and Computer Science, 2011, roč. 2011, č. 1, s. 463-468. ISSN: 2078-0958.

MATOUŠEK, R., LANG, S., MINÁŘ, P., PIVOŇKA, P. *Evolutionary Design of Polynomial Controller*. An international Journal of Science, Engineering and Technology, World Academy of Science Engineering and Technology, 2011, roč. 2011, č. 59, s. 639-644. ISSN: 2010-376X.

MATOUŠEK, R., MINÁŘ, P., LANG, S., ŠEDA, M. HC12: *Efficient PID Controller Design*. Engineering Letters, 2012, roč. 20, č. 1, s. 42-48. ISSN: 1816-093X.

LANG, S., ŠVARC, I. *2DOF Controller Tuning*. Advanced in Soft Computing, AISC Springer, s. 68-72, 2017, ISSN: 1803-3814.

9 SEZNAM POUŽITÉ LITERATURY

- [1] MATOUŠEK, R., LANG, S.; MINÁŘ, P.; PIVOŇKA, P. *Evolutionary Design of Polynomial Controller*. An international Journal of Science, Engineering and Technology, World Academy of Science Engineering and Technology, 2011, roč. 2011, č. 59, s. 639-644. ISSN: 2010- 376X.
- [2] ARAKI, M., TAGUCHI, H. *Two-Degree-of-Freedom PID Controllers*. International Journal of Control, Automation and Systems Vol. 1, No. 4, December 2003
- [3] LANG, S. *Optimalizace řídicího algoritmu pomocí evolučního algoritmu*. VUT Brno, 2010, 64 s.
- [4] BLAHA, P.; VAVŘÍN, P. *Řízení a regulace I: Základy regulace lineárních systémů - spojitě a diskrétní*. VUT Brno, 2005, 214 s.
- [5] ŠOLC, F.; VÁCLAVEK, P.; VAVŘÍN, P. *Řízení a regulace II: Analýza a řízení nelineárních systémů*. VUT Brno, 2006, 228 s.
- [6] ŠTECHA, J.; HAVLENA V. *Teorie dynamických systémů*. Ediční středisko ČVUT, Praha, 2005. 248 s.
- [7] PIVOŇKA, P. *Optimalizace regulátorů*. VUT Brno, 2005, 112 s.
- [8] BLAHA, P. *Teorie dynamických systémů: Vnitřní popis dynamických systémů*. VUT Brno, 2008. 49 s.
- [9] ZELINKA, I., OPLATKOVÁ, Z., ŠEDA, M., OŠMERA, P., VČELAŘ, F. *Evoluční výpočetní techniky : Principy a aplikace*. Praha : BEN - technická literatura, 2008. 536 s. ISBN 978-80-7300-218-3
- [10] NĚMEC, Z. *Prostředky automatického řízení: elektrické*. 2. vyd. VUT Brno, 2008.
- [11] ŠVARC, I., ŠEDA, M., VÍTEČKOVÁ, M. *Automatické řízení*. 1. vyd. Brno, CERM., 2007. 324 s. ISBN 978-80-214-3491-2.
- [12] ŠŤASTNÝ, J. *Simulace systémů*. VUT Brno, FSI, ÚAI, 2002. 98 s.
- [13] ŠVARC, Ivan. *Základy automatizace*. VUT Brno, FSI, UAI, 2002. 102 s.
- [14] PIVOŇKA, P. *Číslíková řídicí technika*. VUT Brno, 2003. 151 s.
- [15] WIKIPEDIE. *Wattův odstředivý regulátor* [on-line]. 2004, 21. 8. 2009 [cit. 9.5.2013]. Dostupné z: http://cs.wikipedia.org/wiki/Wattův_odstředivý_regulátor
- [16] BLAHA, P. *Modelování a identifikace: Neparametrické metody identifikace* [on-line]. 2009. 69 s. [cit. 9.5.2013]. Dostupné z: <http://sites.google.com/site/modelovaniidentifikace/prednasky>
- [17] BLAHA, P. *Algebraické a robustní řízení* [on-line]. [cit. 9.5.2013]. Dostupné z: <https://sites.google.com/site/algebraickearobustnirizeni/prednasky>
- [18] MDAT. *Trading Optimization: Simulated Annealing* [on-line]. 2009. [cit. 20.5.2010]. Dostupné z: <http://www.maxdama.com/2008/07/trading-optimization-simulated.html>
- [19] JURA, P. *Signály a systémy: Část 1 Spojitě signály*. VUT Brno, 2003. 81 s.
- [20] JURA, P. *Signály a systémy: Část 2 Spojitě systémy*. VUT Brno, 2003. 78 s.
- [21] JURA, P. *Signály a systémy: Část 3 Diskrétní signály a diskrétní systémy*. VUT Brno, 2003. 89 s.

- [22] CARAMIA, M., DELL'OLMO, P. *Multi-objective Management in Freight Logistics (Multi-objective optimization)*. Herdcover. 2008. ISBN 978-1-84800-381-1
- [23] COELLO, C., LAMONT, G., VELDHIJZEN, D. *Evolutionary Algorithms for Solving Multi-objective Problems*. 2nd edition. 2007. Springer Science+Business Media, LLC. ISBN 978-0-387-33254-3
- [24] MARLER, R., ARORA, J. *Survey of multi-objective optimization methods for engineering*. Springer Verlag. 2004. 27 s. DOI 10.1007/s00158-003-0368-6
- [25] HUBA, M., ŤAPKA, P., ŽILKA, V. *Constrained PID Control*. CRR VUT Brno, 2011, 197 s.
- [26] ÅSTRÖM, K.J., HÄGGLUND, T. *Revisiting the Ziegler–Nichols step response method for PID control*. Journal of process control, Volume 14, Issue 6, September 2004, Pages 635–650
- [27] ÅSTRÖM K. J., HÄGGLUND T. *PID Controllers: Theory, Design, and Tuning*. Research Triangle Park, NC: Instrument Soc. Amer., 1995.
- [28] LEVINE W. S., PISCATAWAY Ed. *The Control Handbook: PID Control*. NJ: IEEE Press, 1996, pp. 198–209.
- [29] Ang, K.H. and Chong, G.C.Y. and Li, Y. (2005) *PID control system analysis, design, and technology*. IEEE Transactions on Control Systems Technology 13(4): pp. 559-576.
- [30] VÍTEČKOVÁ, M., VÍTEČEK, A. *Metoda násobného dominantního pólu pro regulátory se dvěma stupni volnosti a proporcionální soustavy s dopravním zpožděním*. IC Cybernetics and Informatics, Vyšná Boca, 2010.
- [31] OŠMERA, P.; ŠEDA, M.; MATOUŠEK, R.; WEISSER, R. *Two-level Transplant Evolution for Optimization of Controllers*. In 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) (id 19801). 2011. Shanghai: IEEE, 2011. s. 2110-2115. ISBN: 978-1-61284-180-9.
- [32] WEISSER, R.; OŠMERA, P.; ROUPEC, J.; MATOUŠEK, R. *Two-Level Transplant Evolution for Optimization of General Controllers*. In New Trends in Technologies: Control, Management, Computational Intelligence and Network Systems. 2010. Croatia, Sciyo, 2010. s. 50-68. ISBN: 978-953-307-213-5.
- [33] WIKISOFIA. *Různé definice systémů*. 2016, 6. 3. 2016 [cit. 21.3.2014]. Dostupné z: https://wikisofia.cz/wiki/Vysv%C4%9Btlete,_co_je_to_syst%C3%A9m_a_informa%C4%8Dn%C3%AD_syst%C3%A9m_a_popi%C5%A1te_jejich_z%C3%A1kladn%C3%AD_vlastnosti
- [34] WIKIPEDIE. *Zpětná vazba*. [on-line]. 2008, 23. 11. 2013 [cit. 21.3.2014]. Dostupné z: http://cs.wikipedia.org/wiki/Zp%C4%9Bn%C3%A1_vazba
- [35] WEISER, R. *Evoluční optimalizace řídicích algoritmů*. VUT v Brně, 2010. 37 s.
- [36] KUPKA, L. *Nastavení PID regulátorů*. [on-line] TU Liberec, fakulta mechatroniky. 26 s. [cit. 21.3.2014]. Dostupné z: http://www.fm.tul.cz/~libor.kupka/09_-_Nastavovani_PID.pdf
- [37] JURA, P. *Fuzzy logika v modelování a řízení* [prezentace]. VUT v Brně. 41 s.

- [38]BŘEZINA, T. *Efektivní metoda Q-učení: Simulační posouzení použitelnosti pro řízení aktivního magnetického ložiska*. VUTIUM. VUT v Brně, 2003. ISBN: 80-214-2414-1, ISSN 1213-418X
- [39]MATOUŠEK, R. *Pokročilé metody počítačové inteligence*. VUTIUM. VUT v Brně, 2012. ISBN: 978-80-214-4341-0, ISSN 1213-418X
- [40]Dreaming of metaheuristics. *Classification of metaheuristics*. 12.10.2007. [cit. 28.3.2014].
Dostupné z: <http://nojhan.free.fr/metah/index.php?q=differential+evolution>
- [41]KIRKPATRICK, S., GELATT, C. D., VECCHI, M. P. *Optimization by Simulated Annealing*. Science, New Series, Vol. 220, No. 4598. (May 13, 1983), str. 671-680
- [42]KENNEDY, J., EBERHART, R. *Particle Swarm Optimization*. Proceedings of IEEE International Conference on Neural Networks IV., 1995, str. 1942-1948.
- [43]WIKIPEDIE. *Evolutionary algorithm* [on-line]. 2. 4. 2003, 31.3.2014 [cit. 19.4.2014]. Dostupné z: http://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms
- [44]SHARK PROJECT. *EALib Documentation*. [on-line]. [cit. 19.4.2014]. Dostupné z: <http://shark-project.sourceforge.net/2.1.2/doc/EALib/index.html>
- [45]SYCHRA, O., KLIMEŠ, J., ŠIROKÝ, P. *Evoluční biologie*. [on-line]. 2012 [cit. 19.4.2014].
Dostupné z: http://www.zoologie.frasma.cz/obecn%C3%A1%20zoologie/obecna_zoo.html
- [46]WIKIPEDIE. *Genetic algorithm* [on-line]. 16. 1. 2010, 15.4.2014 [cit. 19.4.2014]. Dostupné z: http://en.wikipedia.org/wiki/Genetic_algorithm
- [47] WIKIPEDIE. *Genetic programming* [on-line]. 16. 1. 2005, 16.1.2014 [cit. 19.4.2014]. Dostupné z: http://en.wikipedia.org/wiki/Genetic_programming
- [48]WIKIPEDIE. *Backus-Naur form*. [on-line]. 1. 10. 2004, 22.4.2014 [cit. 19.4.2014]. Dostupné z: http://en.wikipedia.org/wiki/Backus-Naur_form
- [49]WIKIPEDIE. *Creuament (algorisme genètic)*. [on-line]. 15. 6. 2006, 22.4.2014 [cit. 9.5.2014].
Dostupné z: [http://ca.wikipedia.org/wiki/Creuament_\(algorisme_gen%C3%A8tic\)](http://ca.wikipedia.org/wiki/Creuament_(algorisme_gen%C3%A8tic))
- [50]Metaheuristics for Optimal Transfer of P2P Information in VANETs. *Genetic Algorithm (GA)*. [on-line]. 2009. Dostupné z: <http://neo.lcc.uma.es/staff/jamal/portal/?q=content/genetic-algorithm-ga>
- [51]WIKIPEDIE. *PID regulátor* [on-line]. 2009, 1. 7. 2017 [cit. 21.8.2017]. Dostupné z: https://cs.wikipedia.org/wiki/PID_regul%C3%A1tor
- [52]NARZISI, G. *Multi-Objective Optimization* [prezentace]. New York University, Courant Institute of Mathematical Science, 2008. Dostupné z: cims.nyu.edu/~gn387/glp/lec1.pdf
- [53]TŮMA F. *Teorie řízení*. *Západočeská univerzita*. [online] 2017 [cit. 21.8.2017] Dostupné z: <http://www.kky.zcu.cz/uploads/courses/tr/gmk-tabulka.jpg>
- [54]Mathworks *Matlab&Simulink*. [on-line]. Dostupné z: <http://www.mathworks.com>