



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

ZKRÁCENÁ VERZE DIZERTAČNÍ PRÁCE

SHORTENED VERSION OF 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. 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. 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. 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 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, paretooptimal solution, Paret curve, Paret boundary, metaheuristics, evolutionary algorithm, genetic algorithm, GA, GAR, PID, 2DOF, ITAE

LANG, S. *Řešení spojitých systémů evolučními výpočetními technikami: Zkrácená verze dizertační práce.* Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2017. 32 s. Vedoucí dizertační práce doc. Ing. Radomil Matoušek, Ph.D..

Obsah:

1	Úvod.....	5
2	Teoretický úvod.....	6
2.1	Systém.....	6
2.2	Regulace a regulátory.....	6
2.3	Kvalita regulace.....	7
2.4	Využití metaheuristik v optimalizaci.....	8
2.5	Evoluční algoritmy.....	10
2.6	Implementace GA v prostředí Matlab.....	12
2.7	Více-kriteriální optimalizace a Paretovo optimum.....	13
2.8	Testovací soustavy.....	14
3	Využití EVT při návrhu regulátorů.....	15
3.1	Optimalizace parametrů PID s využitím GA.....	15
3.2	Pareto-optimalizace parametrů PID s využitím GA.....	16
3.2.1	Výsledky testování.....	17
4	Syntéza a optimalizace lineárních regulátorů.....	19
4.1	Klasické a moderní metody syntézy a optimalizace lineárních regulátorů.....	19
4.2	Vlastní metoda pro syntézu a optimalizaci lineárních regulátorů.....	20
4.2.1	Kriteriální funkce.....	20
4.2.2	Simulační model.....	20
4.2.3	Průběh optimalizačního procesu.....	20
4.2.4	Forma a princip kódování lineárních regulátorů.....	21
4.2.5	Úprava zesílení lineárního regulátoru.....	24
4.2.6	Výsledky syntézy GAR pro vybrané soustavy.....	26
4.2.7	Výsledky syntézy GAR pro soustavu s dopravním zpožděním.....	27
5	Závěr.....	29
6	Seznam publikací autora.....	29
7	Seznam použité literatury.....	30
8	Curriculum vitae.....	32

1 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 zkušební 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.

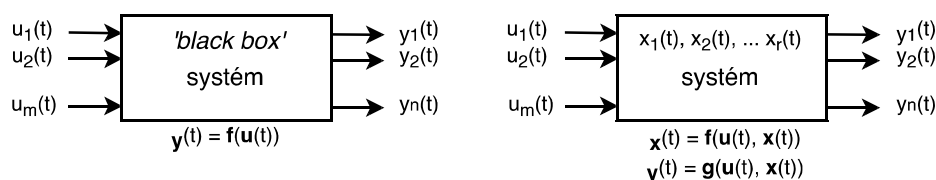
Práce se při návrhu a optimalizaci regulačních obvodů vydává cestou nasazení evolučních výpočetních technik. Je však zároveň demonstrováno, jak zásadní vliv má volba vhodného kódování optimalizačního problému pro efektivitu optimalizace. Vhodné kódování totiž umožní syntézu obecného lineárního regulátoru,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

2 TEORETICKÝ ÚVOD

2.1 Systém

„Systém je uspořádaná množina prvků a vztahů mezi nimi, které z určitého souboru tvoří relativní celek.“ [33] Popis systému lze provádět různými způsoby, z nichž nejjobecnější 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]

V práci budeme pracovat zejména s vnějším popisem, který popisuje systém pouze pomocí vztahů mezi jeho vstupy a výstupy. Výhodou je vyšší stupeň abstrakce. V důsledku ztráty informací a absence znalosti vnitřních stavů není možné řešení systémů s nenulovými počátečními podmínkami.

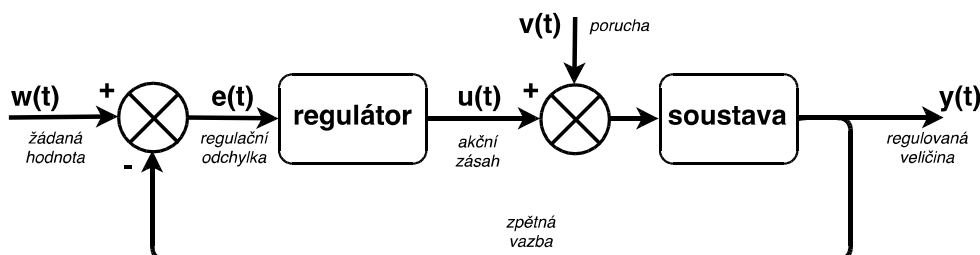


Obr. 1 Vnější a vnitřní popis systému

Základní vlastností lineárních systémů je platnost principu *superpozice*. Lineární systémy mají pouze jeden ustálený stav. [5, 11]. Řešení lineárních systémů je výrazně jednodušší a existuje celá řada popisů alternativních k popisu diferenciálními rovnicemi, 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.

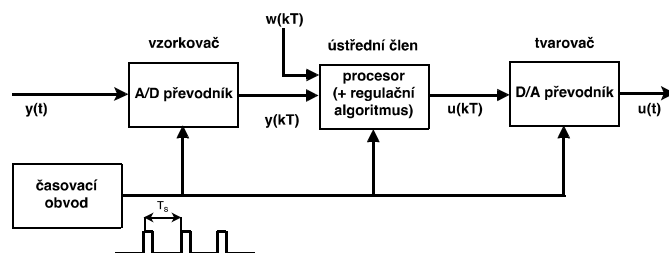
2.2 Regulace a regulátory

Základním prvkem regulace je zpětná vazba, tj. informace o skutečné hodnotě regulované veličiny. „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]. V průmyslu tvoří regulace 85% veškerého řízení [14, 15]. 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). Zbýlých 9% pak tvoří vyšší formy řízení [14, 15].



Obr. 2 Regulační obvod

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 pracovali ve spojitém čase. 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. 3 Číslicový regulátor

Regulátory můžeme členit na základě fyzikálního principu, na němž jsou založeny (regulátory mechanické, hydraulické, pneumatické a elektronické, nebo též hybridní regulátory). Z pohledu teorie řízení je ale zajímavější princip fungování regulátoru a jeho vnitřní struktura, než konečná fyzická realizace. Nelze proto obecně stanovit, který z regulátorů je více či méně vhodný, zda PID či fuzzy nebo dvoupolohová regulace, takové prohlášení můžeme učinit až na základě zhodnocení vlastností regulované soustavy. Uvedme tedy nejvýznamnější skupiny.

- Dvoupolohová regulace
- Regulátory PID
- Obecné lineární regulátory
- Stavové regulátory
- Fuzzy regulátory
- Neuronové sítě
- Obecné nelineární regulátory
- Optimální regulátory
- Adaptivní regulátory

2.3 Kvalita regulace

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 žádané 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.

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.

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

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

Tabulka 1: Běžně používaná integrální kritéria

Důležité 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.

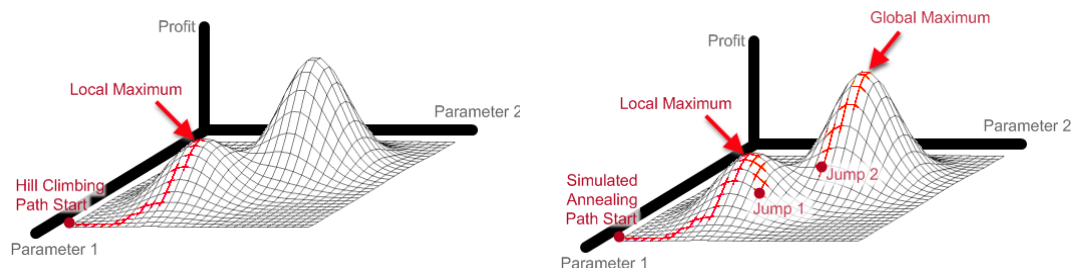
2.4 Využití metaheuristik v optimalizaci

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:

$$\mathbf{x}_{opt} = \arg \text{opt } f(\mathbf{x})$$

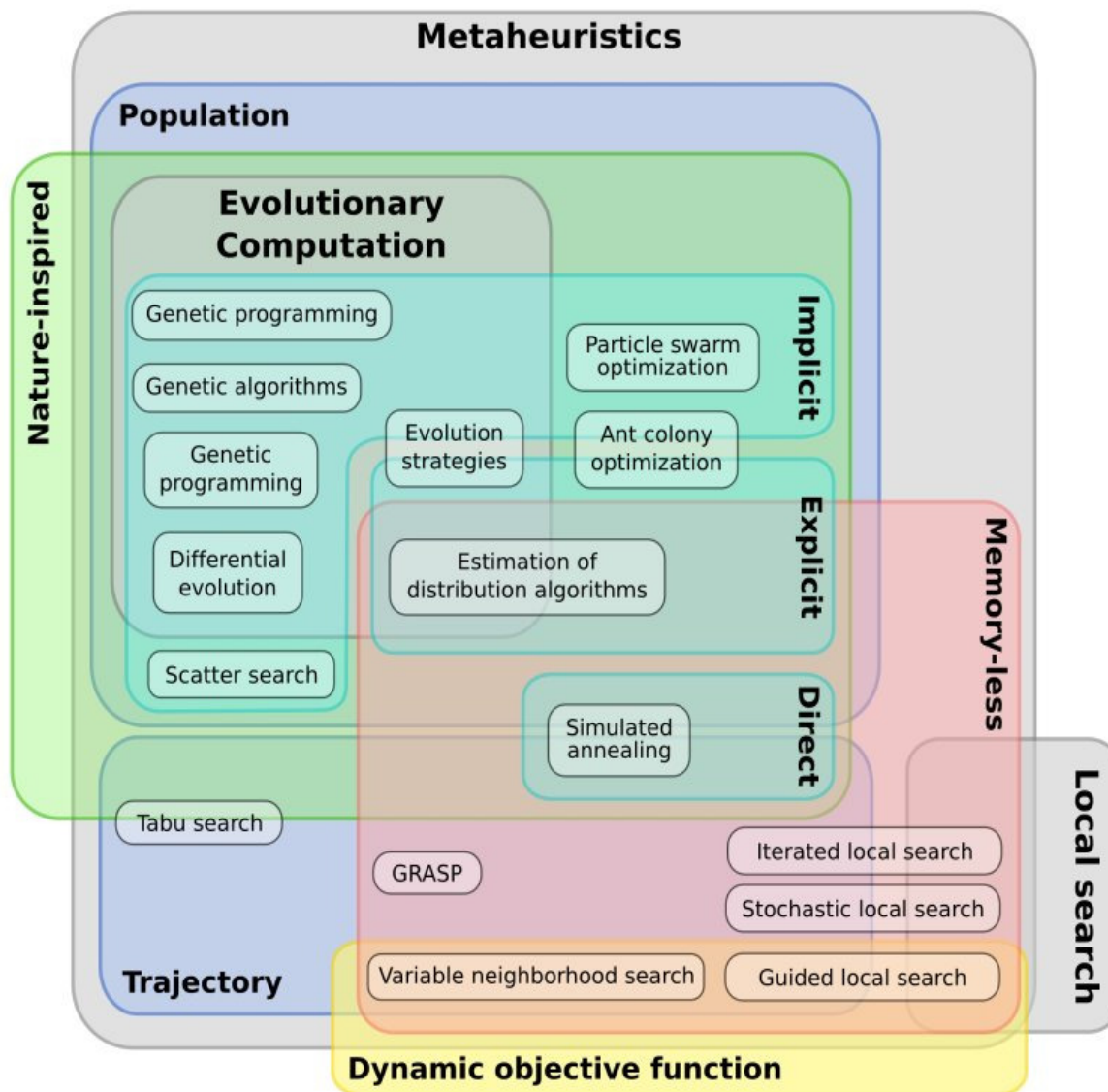
$$\mathbf{x} \in X^m \quad (1)$$

Dostáváme se k řešení matematických úloh infinitesimálního a variačního počtu ve funkcionálních prostorech [25], nebo k nasazení numerických metod. 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. 4 Ilustrace srovnání lokálního a globálního řešiče [18]

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. 5 Možné rozdělení metaheuristik [26]

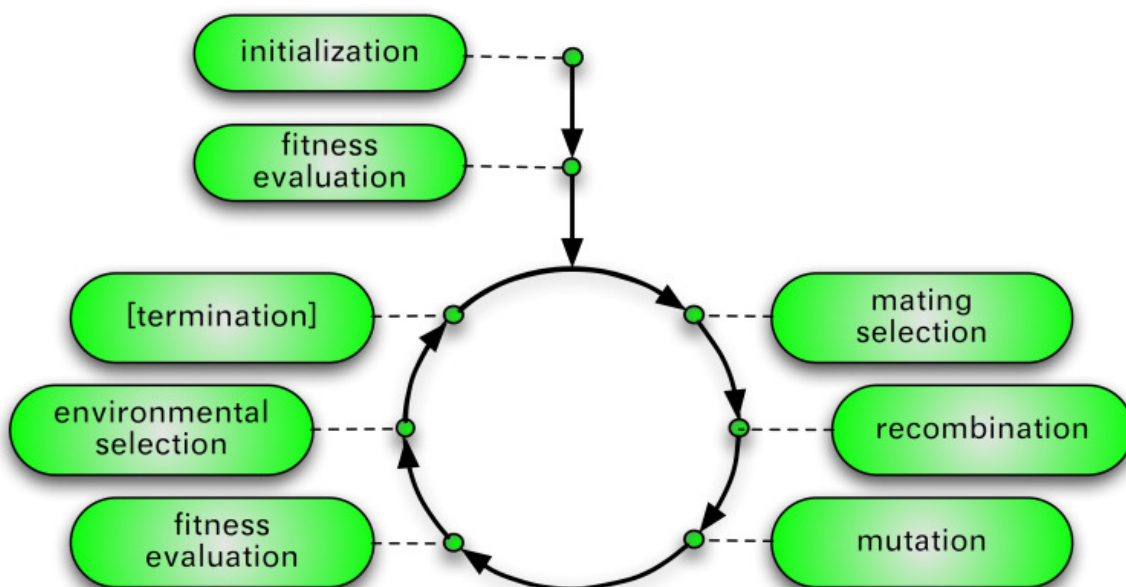
Metodu lokálního hledání (LocalSearch). Průběh optimalizace v rámci jedné iterace je zcela deterministický, neboť zlepšující řešení jsou hledána výhradně ve směru nejvyššího gradientu, nicméně metoda opakuje hledání v rámci iterací z náhodně volených počátečních řešení. Problémem metody je vysoké riziko uvíznutí v lokálním extrému, pokud funkce není monotónní. 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 uvíznutí v lokálním extrému (tentokrát ve formě oscilace mezi dvěma body). V osmdesátých letech minulého století na horolezecký algoritmus navázal F. Glover s **metodou zakázaného prohledávání** (Tabu Search), kdy byl horolezecký algoritmus doplněn o krátkodobou paměť, je tak výrazně sníženo riziko zacyklení.

Z výše uvedené rodiny algoritmů uveďme ještě **metodu simulovaného žhání** (Simulated Annealing). V roce 1983 ji představil S. Kirkpatrick [27], který se nechal inspirovat fyzikálními ději probíhajícími v metalurgii při žhání kovů. Zcela odlišnou kategorií 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. Modifikací metody rojení částic a hejnových algoritmů je celá řada, více [9, 28].

Inspiraci ve fungování živočišných společenstev nalezla i metoda **optimalizace mravenčí kolonií** (Ant colony optimization, ACO), prezentovaná v roce 1992 J. Dorigem, viz [9]. 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 inspiroweme 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].

2.5 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 [16, 17, 35].



Obr. 6 Cyklus evolučního algoritmu [30]

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,

tisíci a milióny 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* [31], 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 selekce jedinců v rámci algoritmů genetické optimalizace.

- **Jedinec** – Pojem *jedinec* označujeme jedno konkrétní řešení optimalizačního problému.
- **Populace** – Množina jedinců, se kterými pracujeme v rámci jednoho optimalizačního cyklu.
- **Generace** – nahrazuje pojem iterace, jedná se o jeden optimalizační cyklus.
- **Fitness** – *Fitness* nebo též *vhodnost* je určením míry kvality jedince.
- **Výběr** – Je-li použito slovo *výběr* samostatně, máme zpravidla na mysli *výběr pro účely křížení*. Metod výběru je celá řada, např. stochastický, poměrový, turnajový, ruletový.
- **Výběr prostředí** – Jedná se o výběr jedinců pro účely dalšího kola optimalizace.
- **Křížení** – je operací, kdy dochází ke kombinaci parametrů původních jedinců a vzniku jedinců nových. Křížení zpravidla zachovává délku genetického řetězce, ale není to podmínkou.
- **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.

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, 29, 32]. 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 [29]. 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. Rechengere 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, 29].

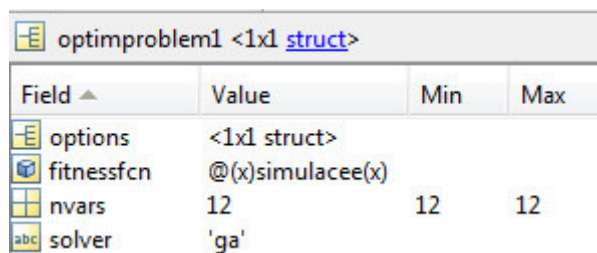
Genetické programování (GP) přechází od optimalizace parametrů, jak tomu je u GA, k optimalizaci struktur. Lawrence J. Fogel v roce 1964 aplikoval evoluční algoritmy k hledání konečných stavových automatů [36], Michael L. Cramer v roce 1985 položil základy moderního pojetí genetické optimalizace založené 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.

Gramatická evoluce (GE) je svým smyslem podkategorií GP, je založena na gramatice v Backus-Naurově formě (BNF, viz [37]), 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.

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, 29]

2.6 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 žíhání, ACO či GA. Nabízí se dvojí 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. V rámci řešení složitějších problémů se jeví výhodnější variantou volání funkcí z kontextu uživatelského programu, jelikož ovládání grafických prvků by bylo pro uživatele zátěží.

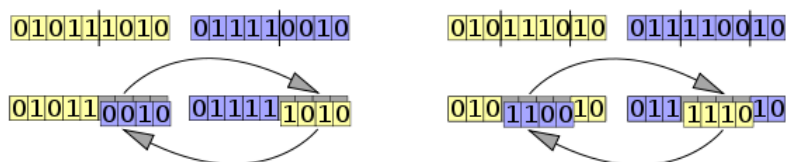


Field	Value	Min	Max
options	<1x1 struct>		
fitnessfcn	@(x)simulacee(x)		
nvars	12	12	12
solver	'ga'		

Obr. 7 Zobrazení základní datové struktury optimalizační úlohy

Dále se zaměříme na nastavení parametrů GA. První skupinou parametrů 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. V rámci nastavení *Selection* je nabízena řada funkcí, z nichž nejvýznamnější jsou stochastická uniformní selekce, selekce typu ruleta a selekce typu turnaj. [9, 39] 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 Gausovská mutace, což znamená, že k jednotlivým prvkům vektorů reprezentujících jedince jsou přičítána náhodná čísla, jejichž pravděpodobnost je dána gausovskou křivkou, jejíž parametry je možné nastavit. Pokud jsou nastavena omezení, je výchozí funkcí adaptivní mutace, více [39]. Mutace je nezbytným prvkem GA, neboť zabraňuje degradaci populace na málo odlišná řešení [9].

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.



Obr. 8 Jednobodové a dvoubodové křížení [38]

Zmíníme ještě tzv. *hybridní funkce*. Jedná se převážně o klasické gradientní funkce. Nalezené řešení pomocí GA totiž 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.

2.7 Více-kriteriální optimalizace a Paretoovo 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 jednokriteriální optimalizační problém popsany jednou kritériální funkcí. Zmíněná oblast optimalizace se v anglické literatuře nazývá povětšinou *single-object optimization*, přičemž optimalizační problém je formulován následovně:

$$\mathbf{x}_{opt} = \arg \min_{\mathbf{x} \in X^m} f(\mathbf{x}) \quad (2)$$

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-kriteriální 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).

Přistupme nyní k pojmu vícekritériální (nebo též multikriteriální) 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 multikriteriální optimalizace již není skalár, ale vektor, jehož složky odpovídají hodnotám dílčích skalárních jednokriteriálních funkcí. Problém lze matematicky formulovat následovně:

$$\mathbf{x}_{opt} = \arg \min_{\mathbf{x} \in X^m} [f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), \dots, f_n(\mathbf{x})] \quad (3)$$

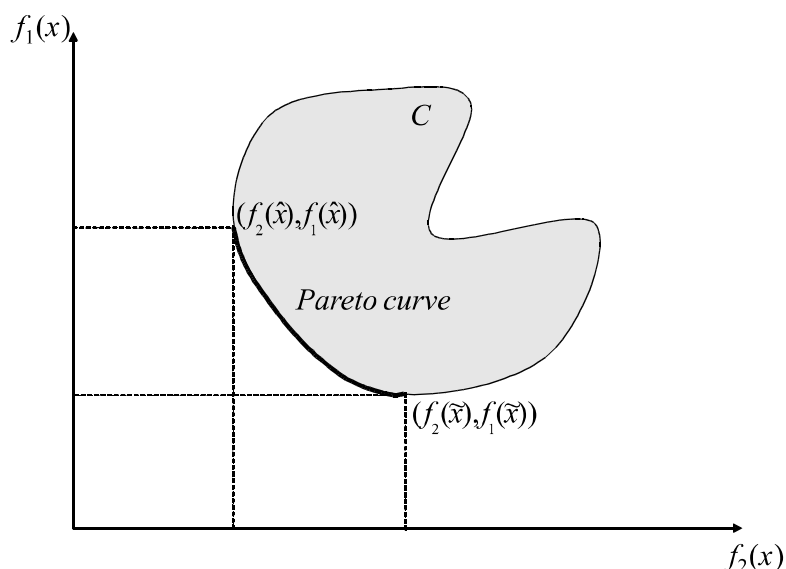
kde $n > 1$ a X je m -dimenzionální prostor přijatelných řešení (více [9, 22]).

V případě jedno-kriteriální optimalizace byla za optimální prohlášena ta řešení, pro něž nabývala kritériální funkce svého globálního extrému v hodnotě. V případě multikriteriální 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.

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]:

Paretoovo optimum: Bod $\mathbf{x}_{opt} \in X$ je pareto-optimální, jestliže neexistuje žádný bod $\mathbf{x} \in X$, pro který platí, že $f(\mathbf{x}) \leq f(\mathbf{x}_{opt})$, a $f_i(\mathbf{x}) < f_i(\mathbf{x}_{opt})$ 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 kritériální funkce. Pareto-optimální řešení tvoří v prostoru řešení množinu diskretních bodů, spojitou či po částech spojitou křivku či hyperplochu. V případě dvoudimenzionálního multikritériálního 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. 9 Paretova hranice [22]

2.8 Testovací soustavy

Pro účely porovnávání výsledků bylo vybráno několik reprezentativních (lineárních) soustav, z nichž některé jsou dále doplněny rovněž o základní nelinearity (saturace, mrtvá zóna, atd.). Ze základních tvarů byla vybrána soustava prvního řádu s dopravním zpožděním ve dvou variantách, V_1 disponuje dopravním zpožděním, které převyšuje svojí 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 zvoleny ale podle publikace [21].

Soustavy V_1 a V_2 jsou popsány následujícími operátorovými přenosy:

$$V_1(s) = \frac{1}{4s+1} e^{-6s} \quad V_2(s) = \frac{1}{4s+1} e^{-1s} \quad (4)(5)$$

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:

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

Druhá varianta soustavy vyššího řádu je označena F_2 , jde o soustavu na mezi stability:

$$F_2(s) = \frac{6}{32s^3 + 16s^2 + 2s + 1} = \frac{0,1875}{(s+0,5)(s+0,25i)(s-0,25i)} \quad (7)$$

3 VYUŽITÍ EVT PŘI NÁVRHU REGULÁTORŮ

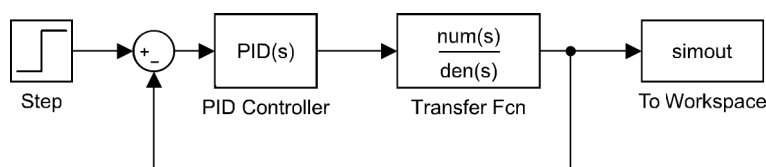
Pro návrh regulátorů resp. pro optimalizaci jejich parametrů byla během historie automatického řízení vyvinuta celá řada algebraických, grafických a empirických metod, a to zejména v průběhu minulého století. Mezi grafické metody patří například návrh tvaru regulátoru v oblasti frekvenčních charakteristik [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.).

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. 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.

3.1 Optimalizace parametrů PID s využitím GA

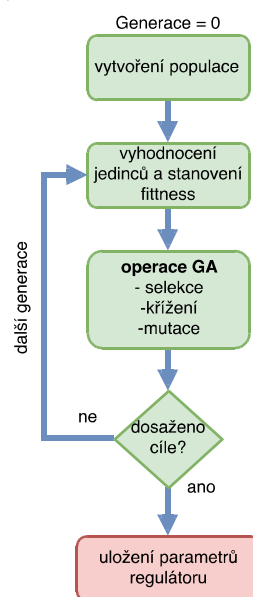
Optimalizační metoda je založena na simulaci. Genetický algoritmus vytváří jednotlivé kombinace parametrů PID, tyto kombinace nahlížíme jako fenotyp jedince v populaci řešení.



Obr. 10 Zjednodušené schéma simulačního modelu

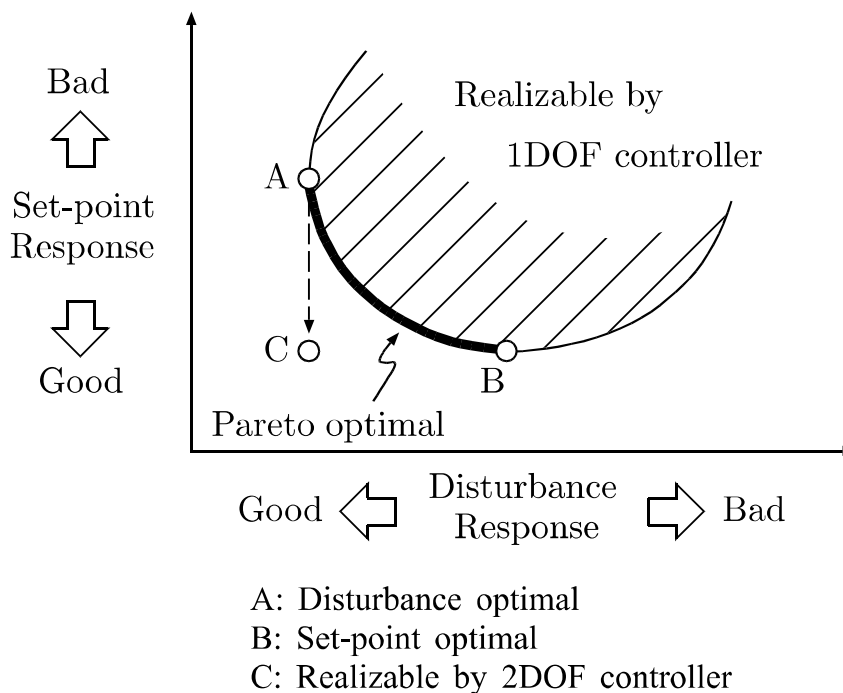
Na schématu je uvedeno zjednodušené schéma regulačního obvodu, jedná se zpětnovazební regulační smyčku obsahující PID regulátor (s filtrací derivační složky) a regulovanou soustavu. Parametry regulátoru jsou nastavovány genetickým algoritmem. Výsledky simulace (časové průběhy) jsou exportovány do workspace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

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.



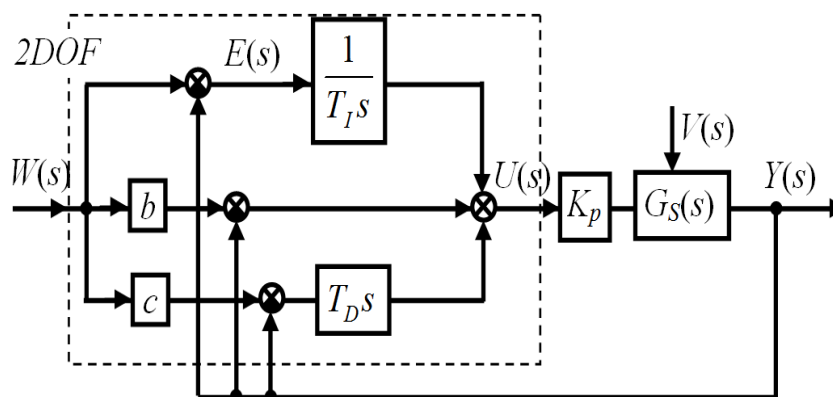
3.2 Pareto-optimalizace parametrů PID s využitím GA

Pareto-optimalizace regulačních úloh je jedním z velmi zajímavých a perspektivních přístupů. Uvedeným tématem se zabývá řada publikací. Za mimořádně zajímavou a kvalitní považuji 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. 11 Ilustrativní příklad pareto-optimalizace regulátoru [2]

Jsou zde hodnocena dvě kritéria, jednak odezva na řízení a odezva na poruchu. 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-hranice PID regulátoru vypadat odlišně. A dále, že ani 2DOF PID často nebude mít jediné globálně optimální řešení, ale více vzájemně nedominantních řešení.

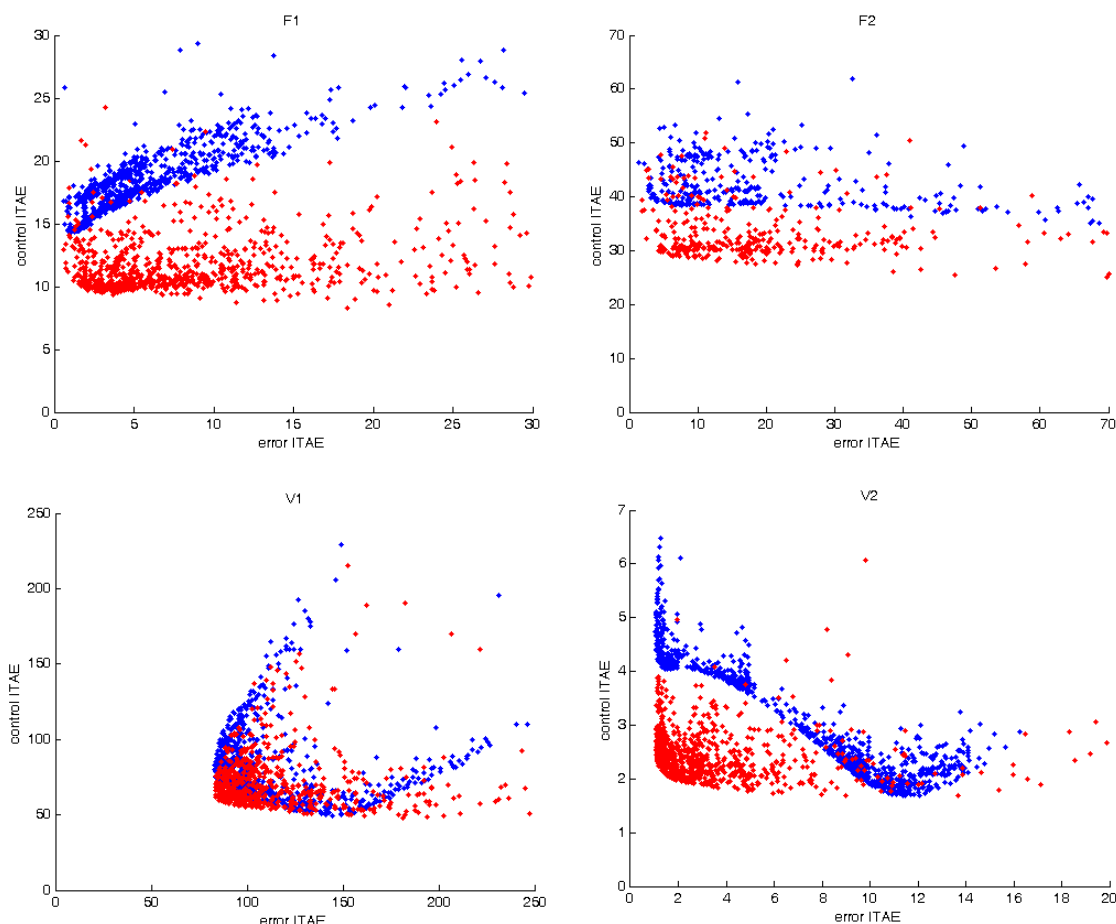


Obr. 12 PID regulátor ve variantě 2DOF

3.2.1 Výsledky testování

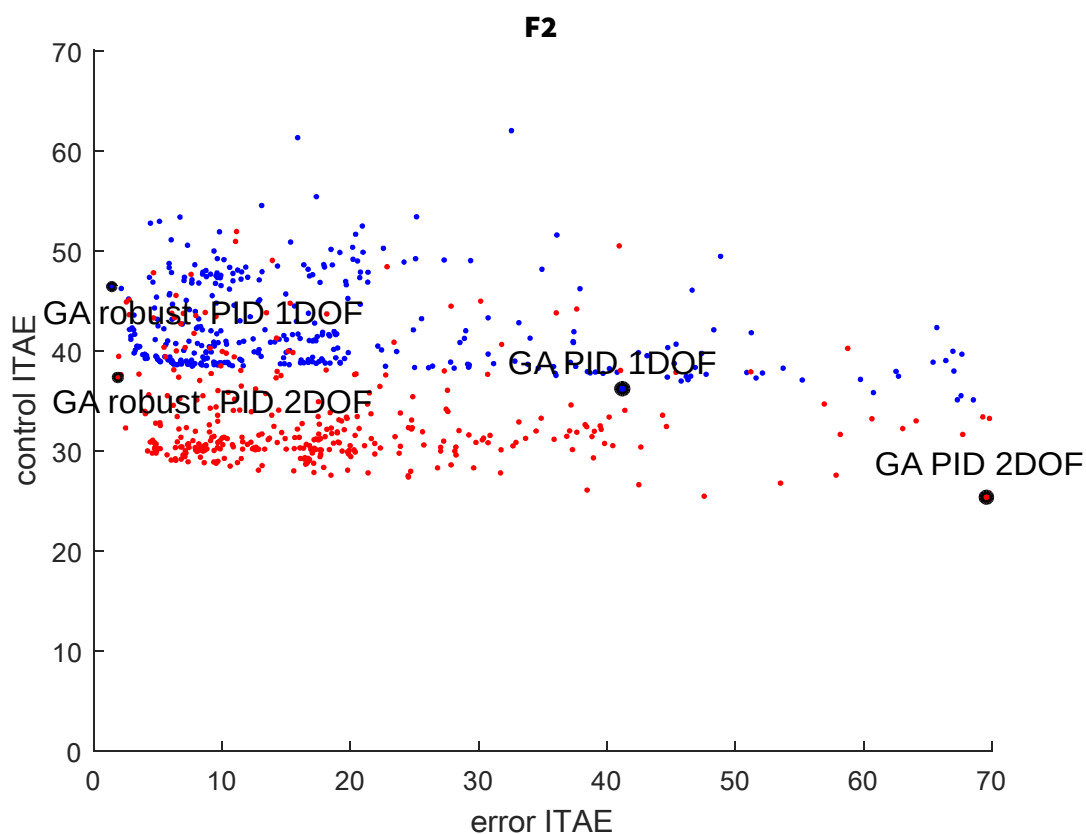
Za účelem otestování a zhodnocení teoretických závěrů bylo již provedeno mnoho testů. Vycházíme ze stejného pojetí dvou-kritériálního posuzování (jako autoři [2]), a sice, že budeme posuzovat kvalitu odezvy na řízení a na poruchu. Jako hodnotící funkce bylo zvoleno integrální kritérium ITAE pro hodnocení 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 paretoovská optima.

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

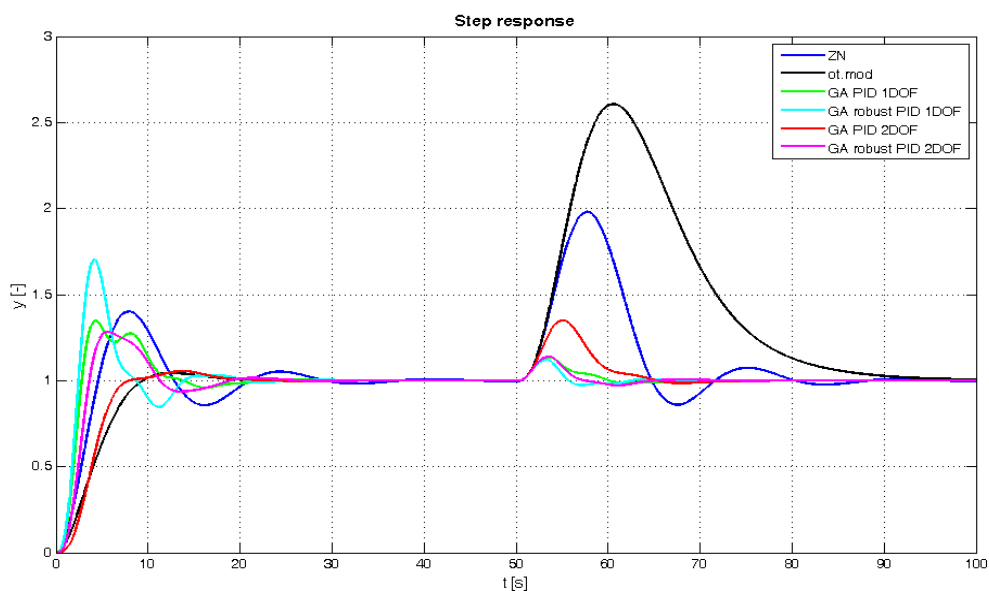


Obr. 13-16 Výsledky multikritériální optimalizace pro soustavy o přenosech F_1, F_2, V_1, V_2

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.



Obr. 17 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. 18 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. 17)

4 SYNTÉZA A OPTIMALIZACE LINEÁRNÍCH REGULÁTORŮ

4.1 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 [10, 14], 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.

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].

$$G_R(s) = \frac{b_m s^m + \dots + b_2 s^2 + b_1 s + b_0}{a_n s^n + \dots + a_2 s^2 + a_1 s + a_0}, m \leq n \quad (8)$$

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 metodou geometrického místa kořene, můžeme daného cíle dosáhnout.

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 umísťování pólů a nul do komplexní roviny. Jestliže totiž přidáme do komplexní roviny kořen, změním tak tvar přenosu, pozice kořenu pak ovlivní i výsledné parametry regulátoru.

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.

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].

$$G_R(s) = K \frac{(s - z_1)(s - z_2) \dots (s - z_m)}{(s - p_1)(s - p_2) \dots (s - p_n)} \quad (9)$$

4.2 Vlastní metoda pro syntézu a optimalizaci lineárních regulátorů

4.2.1 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:

$$I = f_k(\mathbf{x}) \quad (10)$$

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

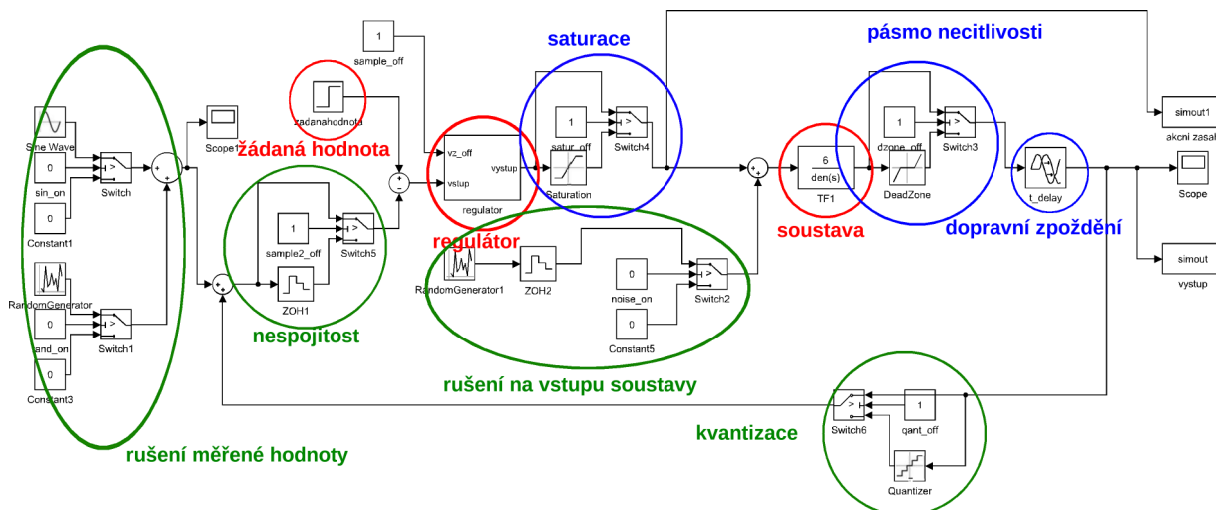
Vektor parametrů regulátoru může nést nejen hodnoty parametrů pevné struktury regulátoru, ale při volbě vhodného kódování může popisovat i samotnou strukturu regulátoru (viz níže).

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).

4.2.2 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. 19 Nastavitelné simulační schéma pro účely optimalizace lineárních regulátorů

4.2.3 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).

4.2.4 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.

4.2.4.1 Verze regulátoru s (výhradně) reálnými kořeny.

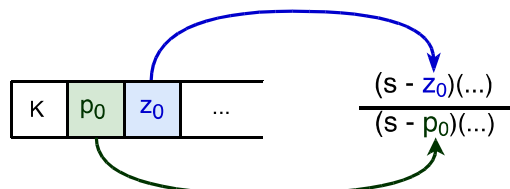
V publikaci [1] tuto variantu pracovně označujeme GAR_n, kde n představuje maximální stupeň polynomu jmenovatele. Stupeň polynomu čitatele je rovný nebo nižší než stupeň polynomu jmenovatele. Pokud během syntézy nastane situace, že polynom čitatele je větší než polynom jmenovatele, pak je polynom ve 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:

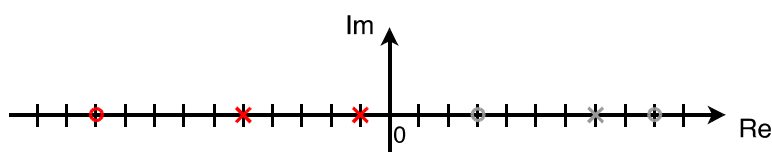
- a) Na začátku genetického řetězce se nalézá gen kódující zesílení přenosu. Hodnota zesílení je optimalizována pouze v kladných hodnotách.



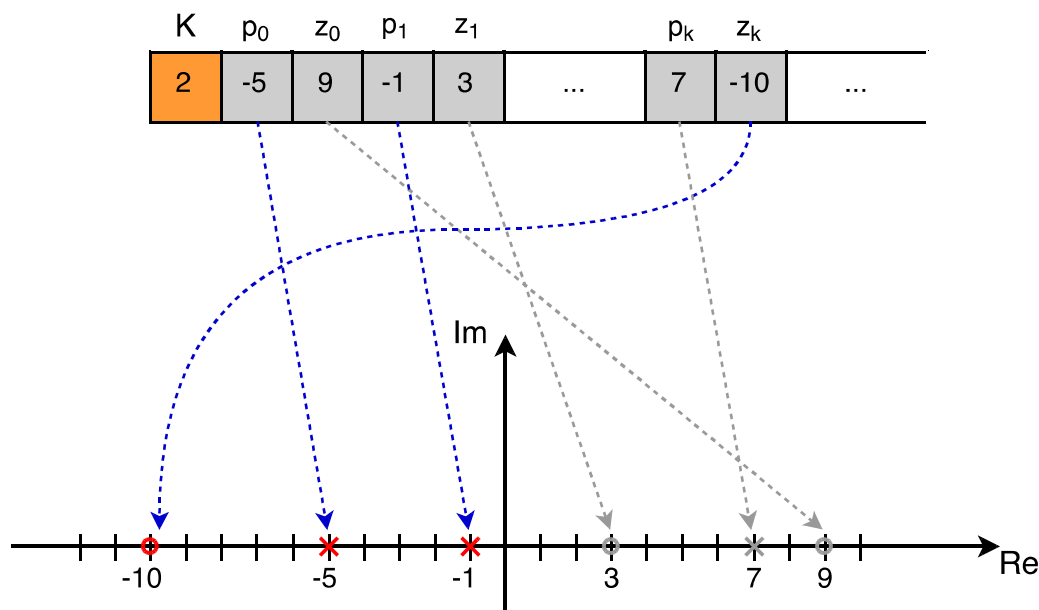
- b) Od druhé pozice jsou v genetickém řetězci kódovány jednotlivé kořeny. Je potřeba odlišit nuly a póly (tj. kořeny náležící čitateli a kořeny náležící jmenovateli). V našem případě bylo zvoleno alternující uspořádání, kdy se v řetězci střídá vždy jeden pól a jedna nula. Toto uspořádání je výhodnější 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. 20 Grafická interpretace kódování/dekódování regulátoru typu GAR do/z řetězce GA.

4.2.4.2 Obecná verze regulátoru s komplexními kořeny

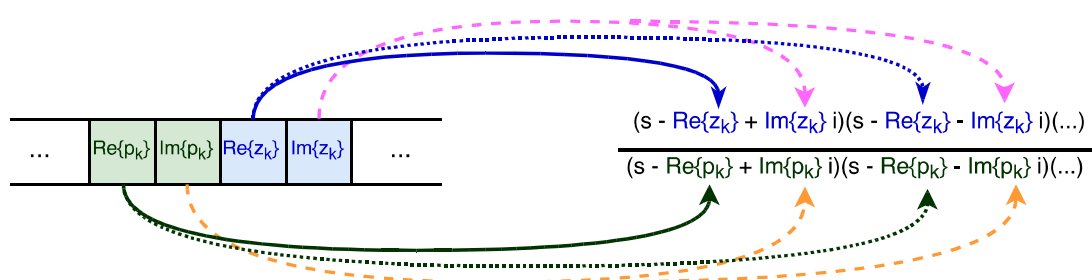
1. Obecnou verzi lineárního regulátoru s komplexními kořeny jsme v publikaci [1] pracovně označili GARNC. Komplexní kořen již není možno kódovat pouze jedním genem, ale pro kódování kořenu jsou použity vždy dva geny, přičemž první gen kóduje reálnou část kořene a druhý gen kóduje imaginární část kořene. 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:

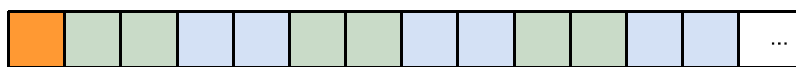
1. Na začátku genetického řetězce leží gen kódující zesílení. Zesílení je optimalizováno pouze v kladných hodnotách.

K	$\text{Re}\{p_0\}$	$\text{Im}\{p_0\}$	$\text{Re}\{z_0\}$	$\text{Im}\{z_0\}$	...
---	--------------------	--------------------	--------------------	--------------------	-----

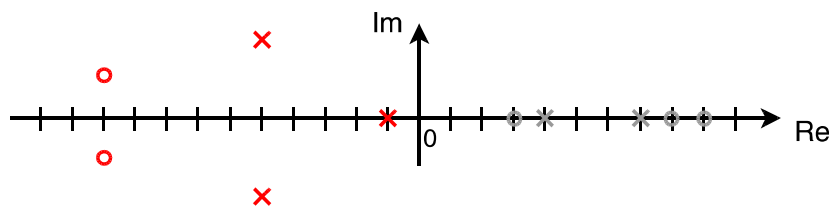
2. Následují dvojice genů kódujících jednotlivé kořeny. První z dvojice genů kóduje reálnou složku, druhý pak imaginární složku kořene.



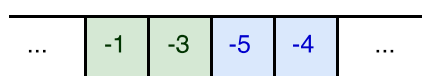
3. V řetězci jsou póly a nuly uspořádány v alternujícím pořadí, tj. vždy jeden pól a jedna nula, následuje další pól a další nula, atd.



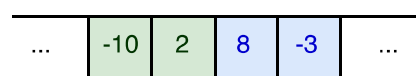
4. Je-li reálná část kořene kladná, kořen se v procesu syntézy neuplatní (imaginární část v tomto procesu eliminace nehraje žádnou roli). Pokud je reálná část nekladná, postupujeme dle následujícího pravidla.



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).

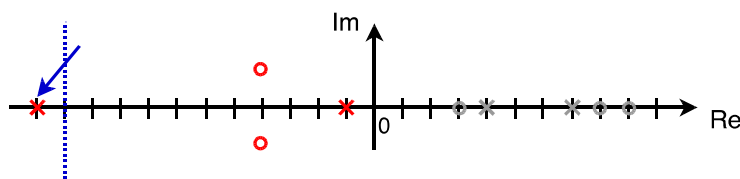


$$\frac{(s+5+4i)(s+5-4i)}{(s+3+i)(s+3-i)}$$

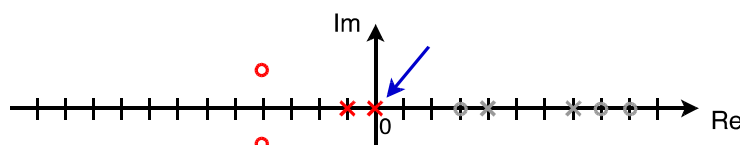


$$\frac{1}{(s+10)}$$

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 umístění 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.

4.2.5 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.

$$k = \frac{b_0}{a_0} = K K_{zp} = K \frac{\prod_{j=1}^m z_j}{\prod_{i=1}^n p_i} \quad (11)$$

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} .

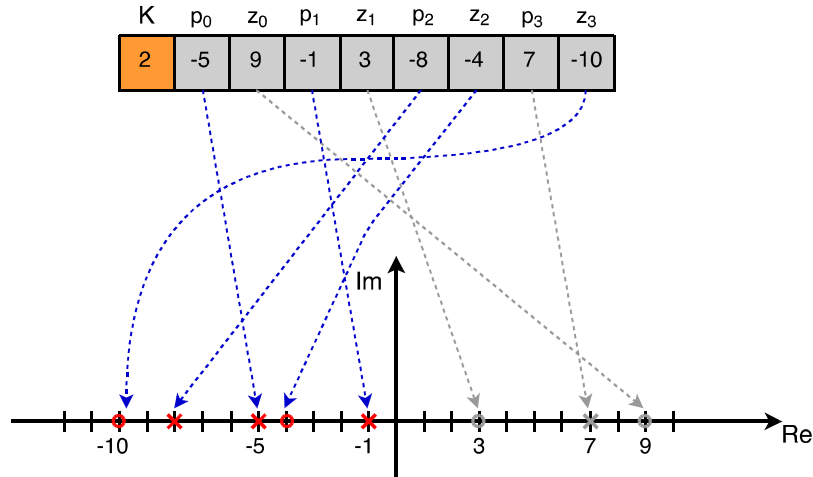
Chceme-li se na provedenou operaci podívat blíže, vyjdeme z nosné myšlenky. Provedme jednoduchou operaci tak, že každý kořenový činitel podělíme absolutní hodnotou jeho kořene a následně všechny takto vytknuté konstanty sloučíme do jediného koeficientu. V podstatě docílíme skutečnosti, že převedeme přenos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

$$G_R(s) = K \frac{(s-z_1)(s-z_2)\dots(s-z_m)}{(s-p_1)(s-p_2)\dots(s-p_n)} = K \frac{\prod_{j=1}^m -z_j \left(\frac{1}{-z_j} s + 1\right) \left(\frac{1}{-z_2} s + 1\right) \dots \left(\frac{1}{-z_m} s + 1\right)}{\prod_{i=1}^n -p_i \left(\frac{1}{-p_1} s + 1\right) \left(\frac{1}{-p_2} s + 1\right) \dots \left(\frac{1}{-p_n} s + 1\right)} \quad (12)$$

$$\dots = K K_{zp} \frac{(t_1 s + 1)(t_2 s + 1) \dots (t_m s + 1)}{(T_1 s + 1)(T_2 s + 1) \dots (T_n s + 1)} = k \frac{(t_1 s + 1)(t_2 s + 1) \dots (t_m s + 1)}{(T_1 s + 1)(T_2 s + 1) \dots (T_n s + 1)}$$

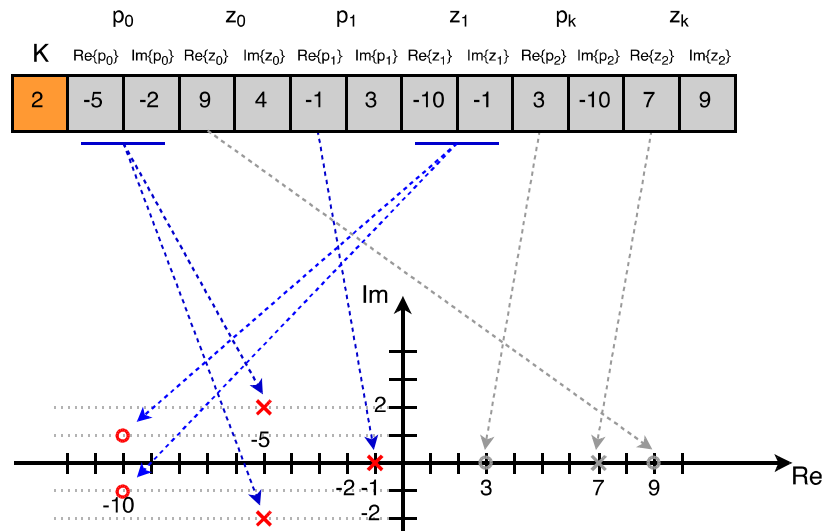
Vhodným krokem je tedy podělit daný vztah právě hodnotou K_{zp} . Praktický dopad bude, že polohou kořenů ovlivňujeme pouze dynamiku, nikoli zesílení, které je optimalizováno samostatně.

$$G_{R\,final}(s) = K \frac{\prod_{i=1}^n \| -p_i \|}{\prod_{j=1}^m \| -z_j \|} \frac{(s-z_1)(s-z_2)\dots(s-z_m)}{(s-p_1)(s-p_2)\dots(s-p_n)} \quad (13)$$



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

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

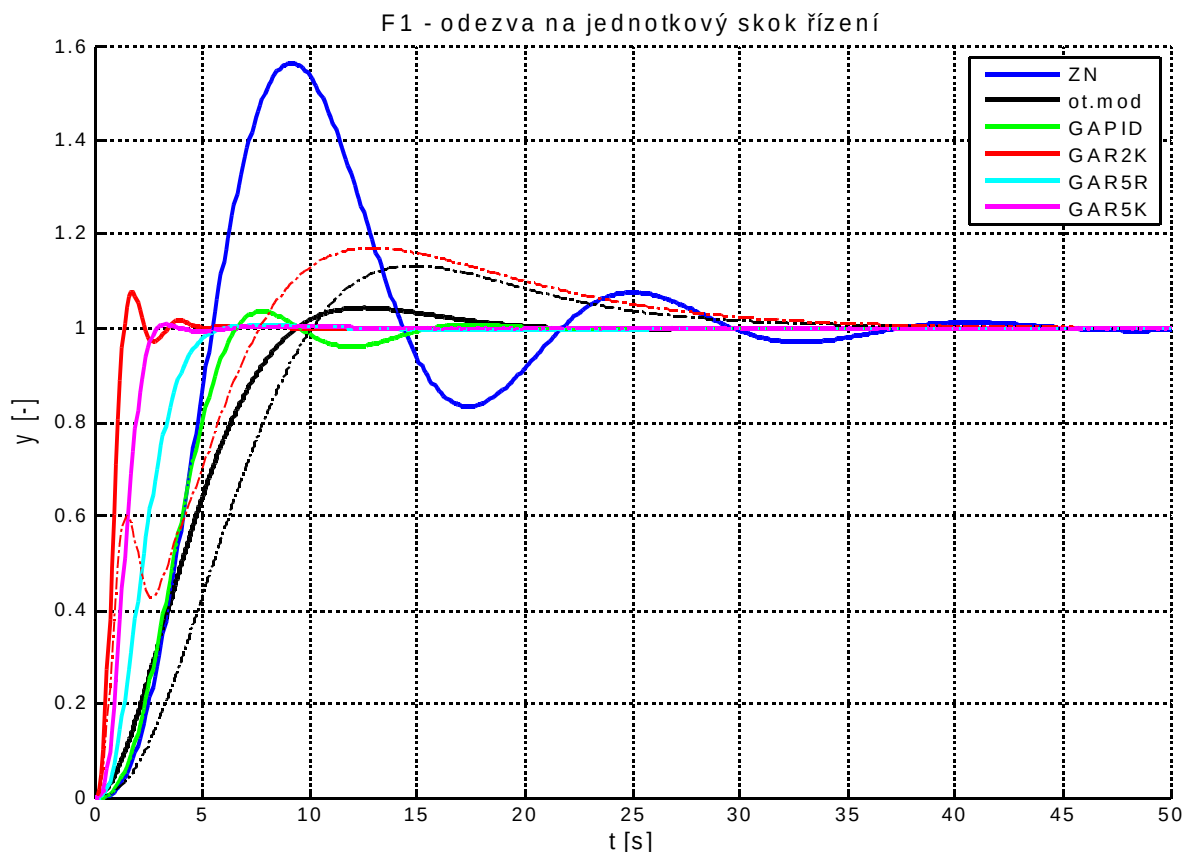


$$G_R(s) = K \frac{\prod_{i=1}^n \| -p_i \|}{\prod_{j=1}^m \| -z_j \|} \frac{(s-z_1)(s-z_2)\dots(s-z_m)}{(s-p_1)(s-p_2)\dots(s-p_n)} = 2 \cdot \frac{2 \cdot \sqrt{5^2 + 2^2} \cdot 1}{2 \cdot \sqrt{10^2 + 1^2}} \frac{(s+10+i)(s+10-i)}{(s+5+2i)(s+5-2i)(s+1)}$$

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

4.2.6 Výsledky syntézy GAR pro vybrané soustavy

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.



Obr. 21 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 seřízené 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)

$$G_{ZN}(s) = \frac{1,58s^2 + 1,00s + 0,16}{0,01s^2 + s} \quad (14)$$

$$G_{optmod}(s) = \frac{s^2 + 0,42s + 0,04}{0,01s^2 + 1s} \quad (15)$$

$$G_{GAPID}(s) = \frac{2,32s^2 + 0,65s + 0,064}{0,1s^2 + s} \quad (16)$$

$$G_{GAR2K}(s) = \frac{122,90s^3 + 116,70s^2 + 32,27s + 2,79}{s^3 + 3,92s^2 + 12,37s} \quad (17)$$

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

$$G_{GAR5K}(s) = \frac{219,90s^4 + 906,00s^3 + 661,20s^2 + 173,40s + 13,86}{s^5 + 10,09s^4 + 46,64s^3 + 114,70s^2 + 118,70s} \quad (19)$$

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.

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 2: Vybrané parametry regulačních obvodů pro soustavu F_1 s navrženými regulátory

4.2.7 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:

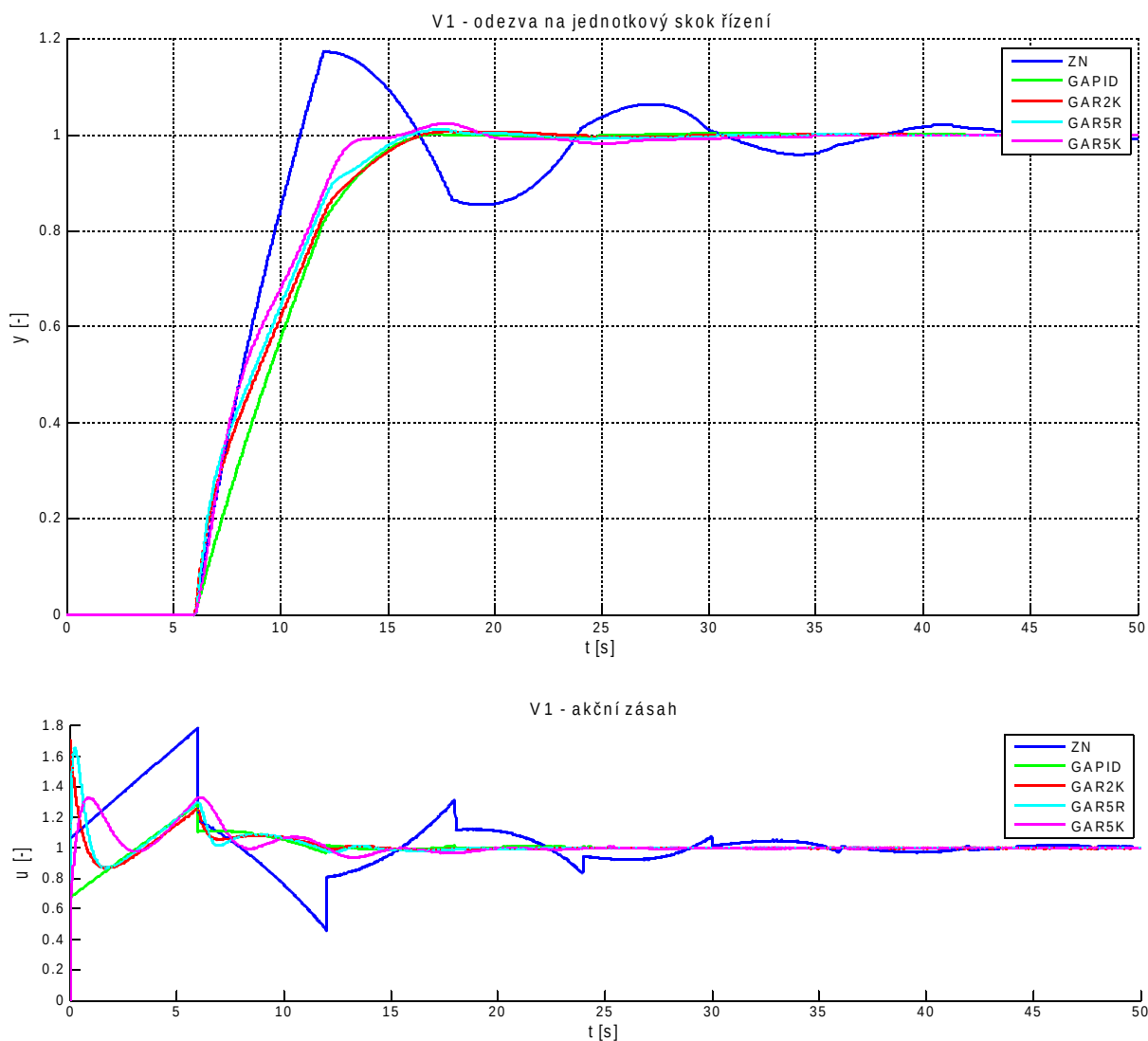
$$G_{ZN}(s) = \frac{2,30p^2 + 1,06p + 0,12}{0,01s^2 + s} \quad (20)$$

$$G_{GAPID}(s) = \frac{1,09s^2 + 0,67s + 0,10}{0,01s^2 + s} \quad (21)$$

$$G_{GAR2K}(s) = \frac{1,72s^2 + 1,09s + 0,17}{s^2 + 1,61s} \quad (22)$$

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

$$G_{GAR5K}(s) = \frac{0,54s^6 + 5,25s^5 + 14s^4 + 17,01s^3 + 10,51s^2 + 3,14s + 0,34}{s^6 + 5,34s^5 + 12,14s^4 + 14,47s^3 + 10,01s^2 + 3,02s} \quad (24)$$



Obr. 22 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 seřazené 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 3: Vybrané parametry regulačního obvodu se soustavou V_1 a navrženými regulátory

5 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 umísťování 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.

6 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.

7 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 - spojité 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] ÅSTRÖM K. J., HÄGGLUND T. *PID Controllers: Theory, Design, and Tuning*. Research Triangle Park, NC: Instrument Soc. Amer., 1995.
- [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] LEVINE W. S., PISCATAWAY Ed. *The Control Handbook: PID Control*. NJ: IEEE Press, 1996, pp. 198–209.
- [15] 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.
- [16] 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.
- [17] 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.
- [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 Spojité signály*. VUT Brno, 2003. 81 s.
- [20] JURA, P. *Signály a systémy: Část 2 Spojité systémy*. VUT Brno, 2003. 78 s.

- [21]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.
- [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., VELDHUIZEN, 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]MATOUŠEK, R. *Pokročilé metody počítačové inteligence*. VUTUM. VUT v Brně, 2012. ISBN: 978-80-214-4341-0, ISSN 1213-418X
- [26]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>
- [27]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
- [28]KENNEDY, J., EBERHART, R. *Particle Swarm Optimization*. Proceedings of IEEE International Conference on Neural Networks IV., 1995, str. 1942–1948.
- [29]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
- [30]SHARK PROJECT. *EALib Documentation*. [on-line]. [cit. 19.4.2014]. Dostupné z: <http://shark-project.sourceforge.net/2.1.2/doc/EALib/index.html>
- [31]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
- [32]WIKIPEDIE. *Genetic algorithm* [on-line]. 16. 1. 2010, 15.4.2014 [cit. 19.4.2014]. Dostupné z: http://en.wikipedia.org/wiki/Genetic_algorithm
- [33]WIKISOFIA. *Různé definice systémů*. 2016, 6. 3. 2016 [cit. 21.3.2014]. Dostupné z: https://wikisofia.cz/wiki/Vysv%C4%Btlete_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%Bt%C3%A1_vazba
- [35]WEISER, R. *Evoluční optimalizace řídicích algoritmů*. VUT v Brně, 2010. 37 s.
- [36]WIKIPEDIE. *Genetic programming* [on-line]. 16. 1. 2005, 16.1.2014 [cit. 19.4.2014]. Dostupné z: http://en.wikipedia.org/wiki/Genetic_programming
- [37]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
- [38]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))
- [39]Mathworks *Matlab&Simulink*. [on-line]. Dostupné z: <http://www.mathworks.com>

8 CURRICULUM VITAE

Osobní údaje

Jména a příjmení: Stanislav Lang
Datum narození: 30.7.1985
Státní příslušnost: česká
Kontakt: 1in.lang@seznam.cz

Vzdělání

PhD. VUT v Brně, Fakulta strojního inženýrství, 2010 – 2017
Dizertační práce: Řešení spojitých systémů evolučními výpočetními technikami
Ing. VUT v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008 – 2011
Diplomová práce: Automatický tester HMI
Ing. VUT v Brně, Fakulta strojního inženýrství, 2008 – 2010
Diplomová práce: Optimalizace řídicího algoritmu pomocí evolučního algoritmu
Bc. VUT v Brně, Fakulta elektrotechniky a komunikačních technologií, 2005 – 2008
Bakalářská práce: Implementace PSD regulátoru do mikrokontroléru MCF52233
– Cyrilometodějské gymnázium, Lerchova 63, Brno, 1997 – 2005

Zahraníční pobyty:

WU Wien, Institute for Statistics and Mathematics, 10/2012 – 01/2013
Technical University of Sofia, Faculty of Machine Technology, 05/2012
Gymnasium Sacre Coeur Wien, 09/2003 – 01/2004

Pedagogická praxe:

1IN (Informatika), cvičení odborného základu, 2010 – 2012
6AA (Automatizace), cvičení odborného základu, 2010 – 2012
VVF (Vyšší formy řízení), laboratorní cvičení, 2011 – 2017

Vedení diplomových prací (ve spolupráci s firmou B+R Automatizace, s r.o.):

Robotický stolní fotbal - herní strategie, Ing. Roman Parák, 2016
Detekce kolize objektů v 3D prostoru, Ing. Jan Grulich, 2015
Robotický stolní fotbal, Ing. Lukáš Jediný, 2015
Robotický stolní fotbal, Ing. Petr Mareček, 2015

Vedení bakalářských prací:

Robotický stolní fotbal - detekce polohy protihráče, Bc. Otrádovský Radek, 2016
Návrh a realizace nové laboratorní úlohy s PLC B&R Automation, Bc. Cejpek Zdeněk, 2016
Návrh a realizace vzorových úloh pro PLC B&R Automation, Bc. Vymazal Aleš, 2016
Návrh a realizace vzorových úloh pro PLC B&R Automation, Bc. Daniel Sýkora, 2013

Pracovní zkušenosti:

VUT v Brně, Fakulta strojního inženýrství, Programátor, 09/2016 – dosud
B+R Automatizace, s r.o., Vývojový pracovník pro robotiku a CNC, 07/2013 – dosud
VUT v Brně, Fakulta strojního inženýrství, Strojírenský technik, 10/2011 – 06/2013
Honeywell International, s r.o., Oddělení pro vývoj domácí automatizace (praxe), 09/2009 – 05/2011
VUES Brno, s r.o., Akreditovaná zkušebna, 08/2007 – 09/2007

Certifikáty:

§7 Vyhláška 50/78 o způsobilosti v elektrotechnice
Návrh měřicích a řídicích systémů v prostředí LabVIEW
Použití mikroprocesorů pro řízení pohonů s BLDC motory
Robustní PID regulátory s omezeními
Němčina pro zaměstnání, Goethe-Institut
Zdravotník zotavovacích akcí, Český červený kříž
Hlavní vedoucí letních táborů s akreditací MŠMT
Nízké lanové překážky s akreditací MŠMT