

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta strojního inženýrství
Ústav mechaniky těles, mechatroniky a biomechaniky

Ing. Roman Weissner

EVOLUČNÍ OPTIMALIZACE ŘÍDICÍCH ALGORITMŮ

Evolutionary Optimization of Control Algorithms

Teze dizertační práce
Ph.D. Thesis

Obor: Inženýrská mechanika

Školitel: prof. Ing. Pavel Ošmera, CSc.

Oponenti: prof. RNDr. Ing. Miloš Šeda, Ph.D.
prof. Ing. Ivan Zelinka, Ph.D.

KLÍČOVÁ SLOVA

umělá inteligence, genetické algoritmy, gramatika, gramatická evoluce, diferenciální evoluce, transplantační evoluce, dvoufázová transplantační evoluce, optimalizace, regulátory

KEYWORDS

artificial intelligence, genetic algorithms, grammatical evolution, differential evolution, transplant evolution, two-level transplant evolution, optimization, controllers

MÍSTO ULOŽENÍ PRÁCE

Knihovna FSI VUT v Brně

Obsah

1	ÚVOD	3
1.1	Cíle práce	4
2	TRANSPLANTAČNÍ EVOLUCE	5
2.1	Použitá gramatika	5
2.2	Inicializace jedince	5
2.3	Křížení - Transplantace	6
2.4	Mutace	7
2.4.1	Pravděpodobnost výběru typu mutace	8
2.5	Pravděpodobnost výběru pravidla	9
3	DVOUFÁZOVÁ TRANSPLANTAČNÍ EVOLUCE	10
3.1	Aritmetická redukce stromu	11
4	META-EVOLUCE	13
5	PRAKTICKÁ ČÁST	14
5.1	Úplná gramatika	14
5.2	Kriteriální funkce	15
5.2.1	Hodnocení simulace v případě chyby	16
5.3	Modifikované schéma DE	18
6	EXPERIMENTÁLNÍ ČÁST	19
6.1	Porovnání regulačních pochodů PSD regulátorů pro různé metody nastavení parametrů	19
6.2	Porovnání regulačních pochodů Takahashiho regulátorů pro různé metody nastavení parametrů	20
6.3	Optimalizace obecných regulátorů	20
6.3.1	Ukázky chybně nastavené optimalizace	21
6.3.2	Ukázky chyb v průběhu simulace	24
6.3.3	Porovnání kriteriálních funkcí	25
6.4	Porovnání obecných regulátorů se setrvačností prvního řádu a PSD regulátorů – kmitavá proporcionální soustava bez dopravního zpoždění	26
7	ZÁVĚR	30
8	LITERATURA	32
9	SEZNAM VLASTNÍCH PUBLIKACÍ	35
	CURRICULUM VITAE	36
10	ABSTRACT	37

1 ÚVOD

V současnosti se teorie řízení dynamických systémů opírá o dobře propracovaný matematický aparát, který efektivně využívá. Při návrhu řídicích struktur se nestudují přímo dynamické systémy, které jsou předmětem řízení, ale jejich vhodně navrhnuté modely. Tvorba modelů vychází ze dvou základních principů. První princip se opírá o sledování chování vstupních a výstupních veličin, bez nutnosti detailně poznat řízený systém, tzv. (*black box*) přístup. Na základě tohoto chování jsme schopni sestavit exaktní model, který vystihuje chování skutečného systému. Druhou možností je proniknout do podstaty systému a na základě znalostí jeho vnitřních zákonitostí vytvořit matematický model. Na popis modelu je možno použít soustavy lineárních, nelineárních diferenciálních, případně diferenčních rovnic, nebo přenosu.

Nejběžnějším nástrojem používaným v aplikacích průmyslového řízení jsou PID regulátory, které jsou používány k regulaci rychlosti, teploty, tlaku a mnoha dalších procesních proměnných. V současnosti se PID regulace využívá ve více než 95 % případů procesního řízení. V posledních letech se používají především mikroprocesorem řízené číslicové regulátory.

Za počátek éry PID regulátorů lze považovat období 1915–1940, kdy vznikaly proslulé regulační firmy Bristol, Fisher, Foxboro, Honeywell, Leeds & Nortrup a Taylor Instrument. Proporcionálně integrační regulátory však byly v průmyslu užívány mnohem dříve. Proporcionální zpětná vazba tvoří základ dobře známého odstředivého regulátoru vynalezeného kolem roku 1750, který byl využit pro řízení otáček větrného mlýnu. Podobný regulátor řídil také otáčky Wattova parního stroje. Na odstředivý regulátor se tehdy pohlíželo jako na jediné zařízení bez rozlišení čidla regulované veličiny, ústředního a akčního členu regulátoru. Následné porozumění významu jednotlivých částí tohoto zařízení bylo klíčovým bodem pro jeho další postupné vylepšování. Konkrétně oddělení čidla od akčního členu umožnilo vývoj hydraulického mechanismu realizujícího proporcionálně–integrační regulátor. První regulátor s derivační složkou byl sestaven v roce 1935.

Prudký rozvoj mikroelektroniky v nedávné době umožnil přechod na číslicové regulátory. To způsobilo další posun ve vývoji regulační techniky a základní řídicí algoritmus průmyslového regulátoru byl doplněn dalšími expertními funkcemi jako je self–tuning, adaptace parametrů regulátoru na změnu okolního prostředí, beznárazové přepínání režimů, apod. V poslední době se začínají uplatňovat také fuzzy regulátory a neuronové sítě.

Při vývoji regulátorů je snaha o zabezpečení vysoké robustnosti a zajištění stability systému, současně s požadavkem na vysokou kvalitu řízení. V současné etapě vývoje neexistuje žádný řídicí algoritmus, který by dokázal splnit všechny formulované požadavky.

Je ovšem třeba konstatovat, že exaktní porovnání i u stejného typu řídicího algoritmu je problematické. Někdy je požadován co nejrychlejší přechodový děj a překmit regulované veličiny nevadí, jindy musí být přechodový děj s minimálním překmitem či zcela bez překmitu. To samozřejmě omezuje použití integrálních kritérií pro vyhodnocení kvality regulačního pochodu. Zejména v reálném provozu, při působení obecných náhodných poruch, může být vyhodnocení nejlepšího nastavení regulátoru velmi obtížné, protože tyto poruchy nejsou zpravidla měřitelné. Navíc reálný systém je velmi často silně nelineární a parametry systému se značně mění s časem nebo např. podle okamžitého výkonu, na kterém systém právě pracuje. Optimální nastavení parametrů řídicího algoritmu je pak zpravidla kompromisem mezi více možnostmi nastavení. Tuto optimalizaci zatím v naprosté většině případů provádí specialista, který má pro tuto specializaci předpoklady. Kvalitní seřízení regulačního obvodu je velmi náročná činnost, kterou je nutné pravidelně opakovat, protože se stárnutím zařízení dochází ke změnám v dynamice procesu.

I když stovky výzkumných ústavů a univerzit na celém světě pracují na vývoji nových adaptivních a inteligentních regulátorů, které by měly pomoci realizovat automatické nastavení parametrů řídicích algoritmů, bylo v průmyslové oblasti dosaženo zatím jen dílčích úspěchů.

Dobře naprogramovaný a nastavený PID regulátor stále zůstává v praxi daleko nejčastěji používaným regulátorem při relativně jednoduchém nastavování. Každá významnější regulační firma

má vypracovány své vlastní varianty řídicích algoritmů a metodiku jejich nastavování. Tyto postupy se vzájemně liší a někdy bývá obtížné najít společnou podstatu, protože algoritmus PID regulátoru je možné realizovat mnoha způsoby. Nejen v české, ale i v cizojazyčné literatuře bývá spojitý PID regulátor často odbyt „knižní“ verzí PID regulátoru a z diskretních tvarů algoritmů bývá uvedena přírůstková verze PSD regulátoru nebo její ekvivalent v Z–transformaci. Spojitý PID regulátor a jeho diskretní tvar ovšem mají řadu dalších variant, které mají mnohem vhodnější vlastnosti pro implementaci v průmyslových procesech, než jaké nabízí základní definice PID regulátoru.

1.1 Cíle práce

Cílem práce je zaměřit se na oblast evolučních metod návrhu řídicích algoritmů.

1. Vybrat a popsat vhodné nástroje pro optimalizaci řídicích algoritmů
2. Optimalizovat řídicí algoritmy obecných regulátorů
3. Vybrat regulovanou soustavu vhodnou pro testování navržených algoritmů
4. Simulačně porovnat vlastnosti evolučně navržených řídicích algoritmů s řídicími algoritmy PSD regulátorů
5. Navrhnout kritériální funkci evolučního algoritmu, která dle zadaných kritérií umožní dosáhnout vysokou kvalitu regulačního pochodu

Klíčovou částí v oblasti optimalizace řídicích rovnic regulátorů pomocí evolučních algoritmů je zejména vytvoření evolučního algoritmu, jenž bude schopen generovat a optimalizovat komplexní struktury řídicích rovnic regulátorů a současně i její parametry.

2 TRANSPLANTAČNÍ EVOLUCE

Pro optimalizaci struktury obecného regulátoru vznikl v této práci nový typ evoluční algoritmu, který byl nazván *transplantační evoluce* (*Transplant Evolution*) (TE). Algoritmus transplantační evoluce vznikl spojením nejlepších vlastností *genetického programování* [13] a *gramatické evoluce* [22, 23, 21, 24, 30, 29, 34, 36, 33].

Tento algoritmus přebírá všechny výhody gramatické evoluce se zpětným zpracováním pravidel gramatiky a přidává navíc další výhodné vlastnosti, kterými jsou např. efektivní změna pravidel gramatiky v průběhu evoluce; nastavení různé pravděpodobnosti výběru pro každé pravidlo gramatiky; použití metody přímého zkracování stromu pomocí algebraických operací; operátory křížení a mutace jsou řízeny pravidly gramatiky; nezávislost na směru zpracování gramatiky.

Jedinci v transplantační evoluci jsou reprezentováni jen fenotypem ve formě stromu. Díky tomu není důležitý postup jeho vytvoření. Genotyp jedince je vytvářen pouze v průběhu inicializace stromu a po jeho vytvoření je zahozen. Křížení a mutace pracuje přímo s fenotypem, za použití pravidel gramatiky. Proces křížení a mutace je tedy ekvivalentní transplantaci orgánů v biologii, při které se rovněž nepracuje na úrovni DNA. Díky této ekvivalenci získala metoda jméno *transplantační evoluce*.

2.1 Použitá gramatika

V ukázkách se předpokládá gramatika uvedena v následujících tabulkách:

$\langle S \rangle$	$::=$	$\langle exp \rangle$
$\langle exp \rangle$	$::=$	$\langle fnc1 \rangle \langle exp \rangle \mid \langle fnc2 \rangle \langle exp \rangle \langle exp \rangle \mid \langle var \rangle \mid \langle const \rangle$
$\langle fnc1 \rangle$	$::=$	$U+ \mid U-$
$\langle fnc2 \rangle$	$::=$	$+ \mid - \mid * \mid /$
$\langle var \rangle$	$::=$	$U_{k-1} \mid U_{k-2} \mid U_{k-3} \mid U_{k-4} \mid U_{k-5} \mid$ $E_k \mid E_{k-1} \mid E_{k-2} \mid E_{k-3} \mid E_{k-4} \mid E_{k-5} \mid$ $dE_k \mid dE_{k-1} \mid dE_{k-2} \mid dE_{k-3} \mid dE_{k-4}$
$\langle const \rangle$	$::=$	$1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \mid 0$

Tabulka 1: Pravidla generativní gramatiky v prefixovém BNF tvaru

$U_{k-1} \mid U_{k-2} \mid U_{k-3} \mid U_{k-4} \mid U_{k-5}$	$::=$	$\langle var \rangle \mid \langle const \rangle$
$E_k \mid E_{k-1} \mid E_{k-2} \mid E_{k-3} \mid E_{k-4} \mid E_{k-5}$	$::=$	$\langle var \rangle \mid \langle const \rangle$
$dE_k \mid dE_{k-1} \mid dE_{k-2} \mid dE_{k-3} \mid dE_{k-4}$	$::=$	$\langle var \rangle \mid \langle const \rangle$
$decimal$	$::=$	$\langle var \rangle \mid \langle const \rangle$
$U+ \mid U-$	$::=$	$\langle fnc1 \rangle$
$+ \mid - \mid * \mid /$	$::=$	$\langle fnc2 \rangle$

Tabulka 2: Pravidla analytické gramatiky pro nestrukturální mutaci

$U+ \mid U-$	$::=$	$\langle var \rangle \mid \langle const \rangle$
$+ \mid - \mid * \mid /$	$::=$	$\langle var \rangle \mid \langle const \rangle$

Tabulka 3: Pravidla analytické gramatiky pro zkracující strukturální mutaci

2.2 Inicializace jedince

Inicializace jedince se provádí náhodným generováním pravidel generativní gramatiky, viz. tabulka 1. Překlad začíná zpracováním počátečního symbolu S a podle pravidel gramatiky je počáteční symbol možné přeložit na libovolný funkční předpis. Postup zpracování gramatiky probíhá

U_{k-1}	U_{k-2}	U_{k-3}	U_{k-4}	U_{k-5}	$::=$	$\langle mfncl \rangle$	$\langle mfncl \rangle$
E_k	E_{k-1}	E_{k-2}	E_{k-3}	E_{k-4}	$::=$	$\langle mfncl \rangle$	$\langle mfncl \rangle$
dE_k	dE_{k-1}	dE_{k-2}	dE_{k-3}	dE_{k-4}	$::=$	$\langle mfncl \rangle$	$\langle mfncl \rangle$
$U_+ \mid U_-$					$::=$	$\langle mfncl \rangle$	

Tabulka 4: Pravidla analytické gramatiky pro prodlužující strukturální mutaci

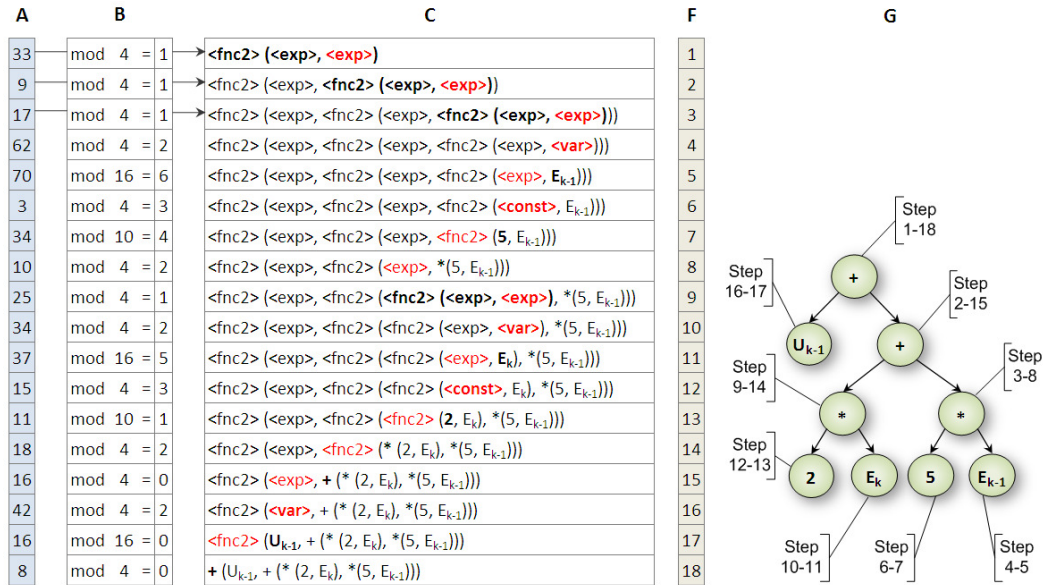
$\langle mfncl \rangle$	$::=$	$\langle fnc1 \rangle \langle var \rangle \mid \langle fnc1 \rangle \langle const \rangle$
$\langle mfncl \rangle$	$::=$	$\langle fnc2 \rangle \langle var \rangle \langle var \rangle \mid \langle fnc2 \rangle \langle const \rangle \langle var \rangle$

Tabulka 5: Rozšiřující pravidla generativní gramatiky

náhodným výběrem, dle vztahu 2.1:

$$random(0, maxInt) \% rules_count \quad (2.1)$$

Kde $random$ je generátor náhodných čísel; $maxInt$ je konstanta; $\%$ značí aritmetickou operaci celočíselného dělení; $rules_count$ značí počet pravidel k výběru.



sloupec A: náhodně generované hodnoty během inicializace stromu, **sloupec B:** výběr pravidla, **sloupec C:** stav překladu. Červené symboly značí neterminály, které budou v dalším kroku nahrazeny, tučně jsou právě nahrazené, **sloupec F:** pořadí operace, **sloupec G:** strom funkce s vyznačeným postupem v jednotlivých krocích (první číslice značí krok ve kterém byl uzel vytvořen, druhé číslo značí krok, kdy byl plně inicializován)

Obrázek 1: Inicializace jedince - zpětné zpracování pravidel

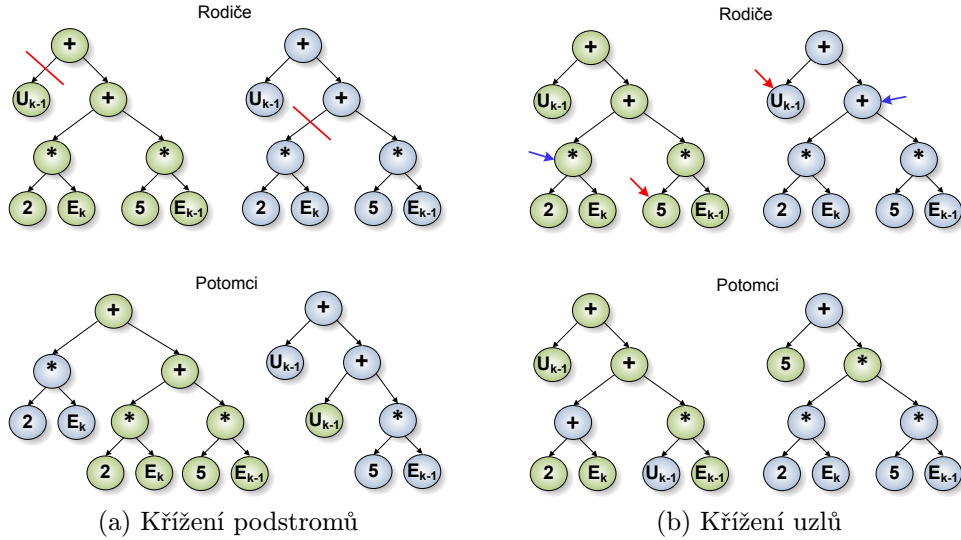
Proces vytváření stromu je ukázán na obrázku 1. Vzhledem k tomu, že každá větev stromu představuje úplně vyčíslitelnou část, tedy takovou, která může stát i samostatně, lze aplikovat křížení, (viz. kapitola 2.3) a mutaci, (viz. kapitola 2.4) přímo na tento vygenerovaný strom. Díky tomu, že transplantační evoluce nepoužívá genotyp pro reprezentaci těla jedince, je možno použít metodu přímého zkracování stromu, která zajistí minimalizaci jeho délky, (viz. kapitola 3.1).

2.3 Křížení - Transplantace

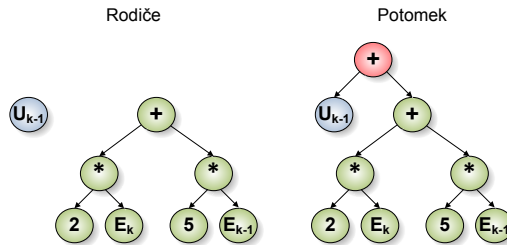
Křížení je metoda zajišťující konvergenci GA a je hlavním způsobem výměny informací mezi jedinci v populaci. Díky tomu, že transplantační evoluce používá genotyp jen pro inicializaci jedince a

dále je řešení reprezentováno pouze fenotypem ve formě stromu, je metoda křížení navržena pro operace se stromovou strukturou, podobně jako v případě genetického programování [13]. Křížení v transplantační evoluci tedy pracuje na vyšší úrovni než v případě jiných evolučních algoritmů, jejichž operace a zásah do fenotypu se děje na úrovni *DNA* a z něj je poté překladem vytvořen fenotyp. V této práci byly navrženy dva typy křížení:

- Křížení částí stromů (z částí dvou rodičů, vznikají dva potomci), viz. obrázek 2,
- Spojování stromů (ze dvou rodičů vzniká jeden potomek), viz. obrázek 3.



Obrázek 2: Křížení částí stromů



Obrázek 3: Spojování stromů

2.4 Mutace

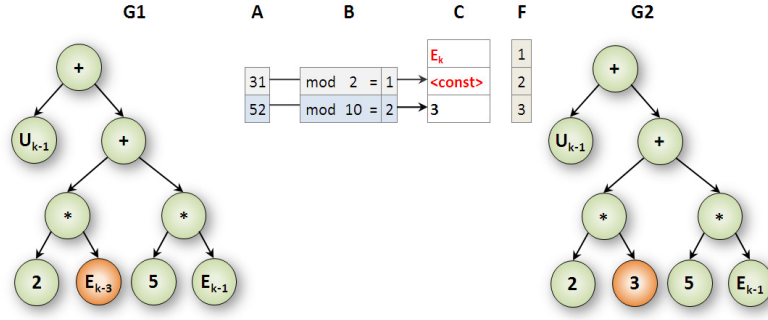
Mutace je jedním z operátorů rekombinace, který svou činností zajišťuje diverzitu populace vkládáním nových znaků do struktury jedince. Mutace je v transplantační evoluci navržena pro operace se stromovou strukturou, tedy s fenotypem jedince, stejně tak jako křížení, (viz. kapitola 2.3).

Proces mutace se v transplantační evoluci provádí použitím pravidel **analytické** G^{-1} a **generativní** G , gramatiky, (viz tabulky 2, 3, 4, 5) a lze jej vyjádřit ve tvaru $\beta_1 \rightarrow \alpha \rightarrow \beta_2$, kde $(\beta_1, \beta_2) \in \Sigma$; $\beta_1 \neq \beta_2$; $\alpha \in N$. To mj. znamená, že terminální symboly před použitím operátoru mutace a po ní jsou vzájemně různé. Jednotlivá pravidla analytické gramatiky mohou mít různou **pravděpodobnost výběru**, stejně tak, jako je tomu v případě generativní gramatiky.

V této práci byly navrženy dva typy mutace, které se dělí podle rozsahu změn které na fenotypu jedince (stromu) způsobí:

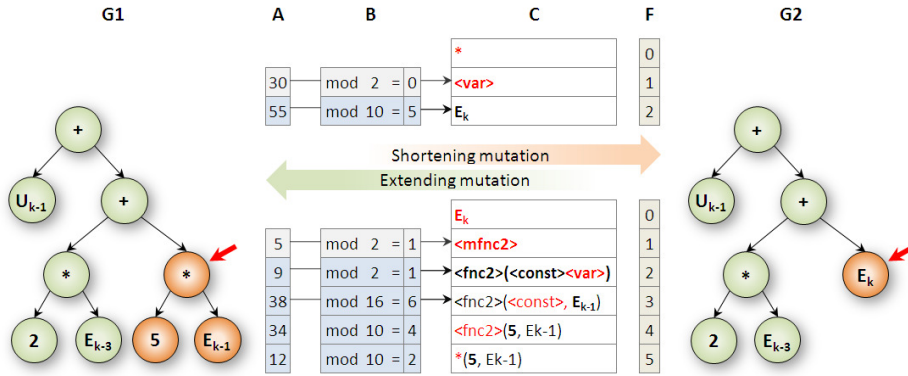
- nestrukturální mutace, viz. obrázek 4,

- strukturální mutace (prodlužující, zkracující), viz. obrázek 5.



sloupec A: náhodně generované hodnoty během mutace, **sloupec B:** výběr pravidla z analytické, nebo generativní gramatiky, **sloupec C:** stav mutace uzlu. Červené symboly značí neterminály, nebo terminály, které budou v dalším kroku nahrazeny, tučně jsou právě nahrazené, **sloupec F:** pořadí operace, **sloupec G1:** strom funkce před použitím operátoru mutace. Uzel k mutaci je vyznačen oranžovou barvou, **sloupec G2:** strom funkce po použití operátoru mutace. Zmutovaný uzel je vyznačen oranžově.

Obrázek 4: Nestrukturální mutace



sloupec A: náhodně generované hodnoty během mutace, **sloupec B:** výběr pravidla z analytické, nebo generativní gramatiky, **sloupec C:** stav mutace uzlu. Červené symboly značí neterminály, nebo terminály, které budou v dalším kroku nahrazeny, tučně jsou právě nahrazené. Nad šipkami je zobrazen výběr pravidel zkracující mutace, pod šipkami je zobrazen výběr pravidel prodlužující mutace, **sloupec F:** pořadí operace, **sloupec G1:** strom funkce před použitím operátoru mutace. Uzel k mutaci je vyznačen oranžovou barvou, **sloupec G2:** strom funkce po použití operátoru mutace. Zmutovaný uzel je vyznačen oranžově.

Obrázek 5: Strukturální mutace

2.4.1 Pravděpodobnost výběru typu mutace

Pravděpodobnost strukturální a nestrukturální mutace se může v průběhu evoluce měnit. Změna pravděpodobnosti může být zapotřebí v případě, kdy cíleným použitím metody *zkracující strukturální mutace*, je možno docílit snížení počtu uzlů ve stromu jedince. Pravděpodobnost použití zkracující strukturální mutace je možno ovlivnit použitím vztahu pro přepočet pravděpodobnosti mutace, (Eq. 2.2):

$$Pc(zm) = \begin{cases} P(zm) & \text{pro } N_c \leq N_m \\ P(zm) \cdot \left(1 + \frac{N_c - M_c}{M_c}\right) & \text{pro } N_c > N_m \end{cases} \quad (2.2)$$

Kde $Pc(zm)$ je přepočítaná pravděpodobnost použití zkracující strukturální mutace; $P(zm)$ je pravděpodobnost použití zkracující strukturální mutace; N_c je počet uzlů obsažených ve stromu; N_m je hranice omezení počtu uzlů

2.5 Pravděpodobnost výběru pravidla

Pro jednoduchost se až dosud ve všech uváděných příkladech předpokládalo, že pravděpodobnost výběru pravidla z množiny pravidel je pro všechny pravidla stejná. Ve skutečnosti tomu tak být nemusí a pravděpodobnost výběru může být pro každé pravidlo jiná a může se dokonce během překladu libovolně měnit. Tímto se transplantační evoluce výrazně liší od všech známých forem evolučních algoritmů založených na gramatice, jelikož změna pravděpodobnosti výběru pravidla by v gramatické evoluci způsobila, že ze stejného genotypu by se vždy vygeneroval jiný fenotyp, což by v nejhorších případech vedlo až k havárii programu¹, viz. např. [22, 21, 23, 24, 26] a další

Nastavení pravděpodobnosti výběru pravidla může být užitečné v případech, kdy je zapotřebí omezit délku stromu a zajistit tak sníženou pravděpodobnost výběru pravidel, které zajišťují větvení stromu. Tato pravděpodobnost se může postupně snižovat s rostoucím počtem uzlů ve stromu. Přepočet pravděpodobnosti výběru pravidel lze realizovat použitím vztahů (2.3), (2.4):

$$Pc(s)_j = \begin{cases} P(s)_j & \text{pro } N_c \leq N_m \\ P(s)_j \left(1 - \frac{N_c - M_c}{M_c}\right) & \text{pro } N_c > N_m \end{cases} \quad (2.3)$$

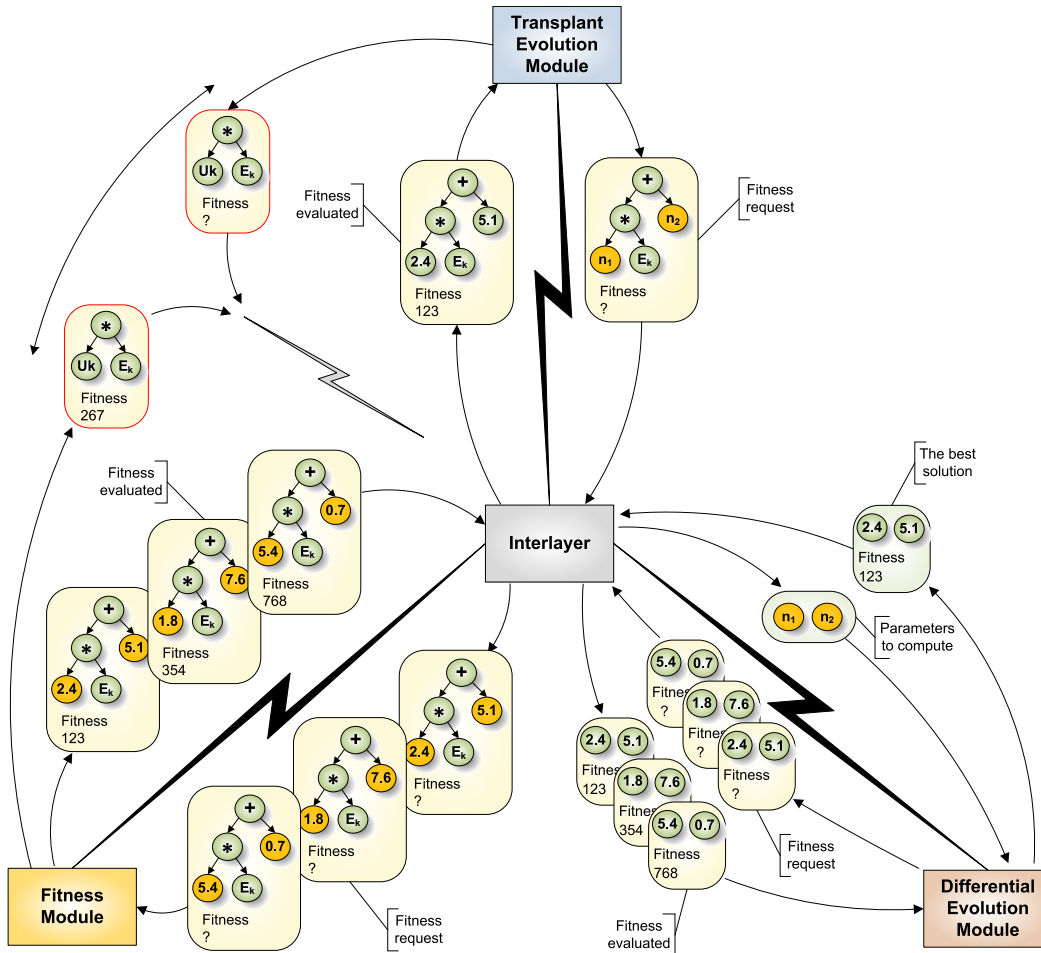
$$Pc(n)_i = \begin{cases} P(n)_i & \text{pro } N_c \leq N_m \\ P(n)_i + \frac{1}{N_j} \left(1 - \sum_{j=1}^{N_j} Pc(s)_j + \sum_{i=1}^{N_i} P(n)_i\right) & \text{pro } N_c > N_m \end{cases} \quad (2.4)$$

Kde: $Pc(s)_j$ vyjadřuje novou hodnotu pravděpodobnosti výběru strukturálního pravidla s indexem j , $P(s)_j$ vyjadřuje přednastavenou hodnotu pravděpodobnosti výběru strukturálního pravidla s indexem j , N_c je počet uzlů obsažených ve stromu, N_m je hranice omezení počtu uzlů, $Pc(n)_i$ vyjadřuje novou hodnotu pravděpodobnosti výběru i -tého nestrukturálního pravidla, $P(n)_i$ označuje přednastavenou hodnotu pravděpodobnosti výběru i -tého nestrukturálního pravidla, N_j je počet strukturálních pravidel, N_i je počet nestrukturálních pravidel.

¹Murphyho zákon: Může-li se něco pokazit, pokazí se to

3 DVOUFÁZOVÁ TRANSPLANTAČNÍ EVOLUCE

Dvoufázová transplantační evoluce, (*Two Level Transplant Evolution*) (TLTE) je novým typem evolučního algoritmu, který vznikl kombinací algoritmu *transplantační evoluce*, (viz. kapitola 2) a *diferenciální evoluce*, viz. [5, 16, 27, 28, 32, 36, 37], případně jakéhokoliv jiného heuristického algoritmu pro optimalizaci numerických parametrů [34, 36, 33]. Spojení dvou evolučních algoritmů má za cíl odstranit obtížné generování parametrů z oboru reálných čísel $\mathbb{R}$, které je diskutováno v [26, 32].



Obrázek 6: Datový tok ve dvoufázové transplantační evoluci s oddělenými moduly

Principem činnosti dvoufázové transplantační evoluce je optimalizace řešení ve dvou fázích. V **první fázi** probíhá optimalizace struktury řešení pomocí transplantační evoluce, poté je tato struktura předána do **druhé fáze**, ve které jsou řešeny její neznámé parametry pomocí diferenciální evoluce, genetického algoritmu, apod. *Diferenciální evoluce* byla vybrána z důvodu rychlosti a schopnosti nalézt řešení, jelikož optimalizace neznámých parametrů rovnice regulátoru je velice náročný a zdoluhavý proces, během kterého dochází k velkému počtu ohodnocení řešení.

Pro zajištění **generování abstraktních numerických parametrů**, je zapotřebí upravit dosud definovanou gramatiku G , viz. tabulky 1, 2, 3, 4, 5. Do této gramatiky je potřeba vložit nové pravidlo, které může být ve tvaru definovaném v tabulce 6. Terminální symbol pravé strany tohoto pravidla je abstraktním numerickým parametrem. Za účelem mutace abstraktních numerických

$$\langle var \rangle ::= \text{num}$$

Tabulka 6: Rozšiřující pravidlo generativní gramatiky

parametrů je potřeba do analytické gramatiky G^{-1} vložit pravidlo, které zajistí přepis terminálního symbolu *num*. Takové pravidlo je uvedeno v tabulce 7.

num	::=	$\langle var \rangle$
-----	-----	-----------------------

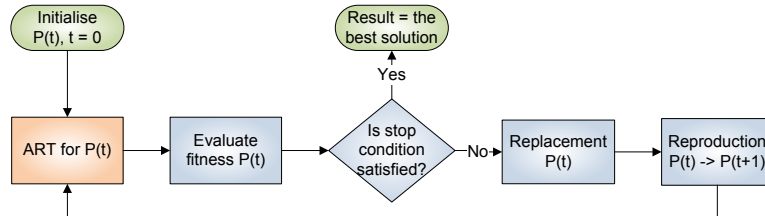
Tabulka 7: Rozšiřující pravidlo analytické gramatiky

Z pohledu implementace je nejnáročnějším procesem propojení dvou evolučních algoritmů. Z důvodu flexibility bylo zvoleno propojení tří samostatných výpočetních modulů, které jsou pro okolí černé skříňky, které komunikují s okolím přes *komunikační interface*. Díky tomuto návrhu lze moduly použít i samostatně pro libovolný jiný optimalizační problém, případně daný modul jednoduše zaměnit za jiný (DE – GA, ...).

Architektura celého řešení a datový tok je ukázán na obrázku a 6.

3.1 Aritmetická redukce stromu

Aritmetická redukce stromu, (*Arithmetic Tree Reducing*) (ART) je metodou, která vyhledává ve stromu jedince takové části, které lze bezztrátově aritmeticky redukovat. V případě, že nalezne takovou část stromu, kterou má obsaženou ve *slovníku*, provede její redukci.



Obrázek 7: Začlenění ART do TE (TLTE)

Metoda je sice použitelná v kombinaci s metodou transplantační evoluce, (viz. kapitola 2), ale její síla se projeví právě ve spojení s dvoufázovou transplantační evolucí, kde je možno využít symbolických konstant v podobě terminálu *num*.

P1	$U+(e)$	::=	e
	$U-(U-(e))$	::=	e
	$U-(U+(e)) \mid U+(U-(e))$	::=	$U-(e)$
P2	$f_{2m}(num, num) \mid f_{2s}(num, num)$	::=	num
P3	$+(a_1, a_2)$	::=	$*(num, a)$
	$-(a_1, a_2)$	::=	num
	$/(a_1, a_2)$	::=	num
P4	$f_{2m}(num, pair(a)) \mid f_{2m}(pair(a), num)$	::=	$*(num, a)$
	$f_{2s}(num, pair(a)) \mid f_{2s}(pair(a), num)$	::=	$*(num, a)$
P5	$f_{2s}(pair(a), pair(a))$	::=	$*(num, a)$

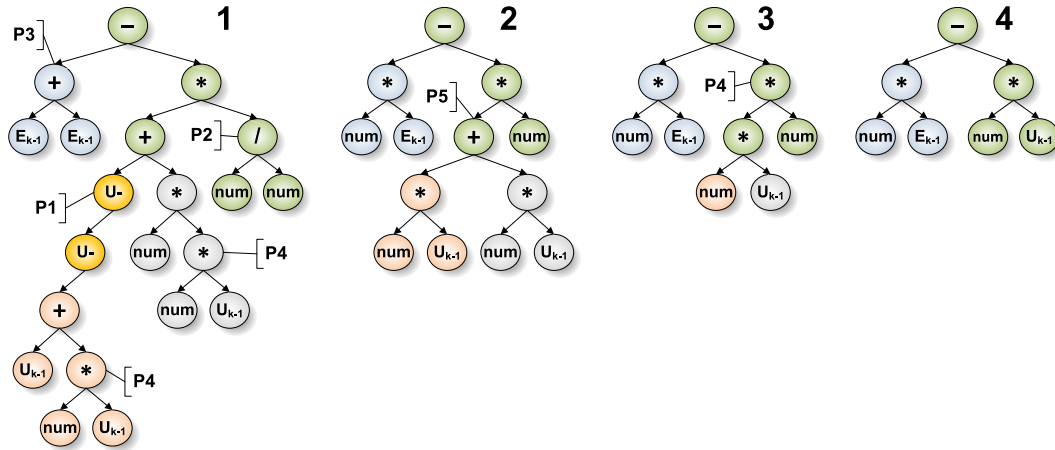
první sloupec: (P1 – P5) slouží pro označení sad pravidel a vyjadřují náročnost dané operace (P1 – nejjednodušší, P5 – nejnáročnější)

Tabulka 8: Pravidla pro ART (základní operace)

Slovník definuje pravidla pro redukční operace se stromem, jenž jsou definována ve tvaru uvedeném v tabulce 8. Pravidla v uvedených tabulkách jsou definována prefixovým zápisem terminálních symbolů, které představují určitý tvar stromu. Význam jednotlivých symbolů je následující: *num* je terminální symbol reprezentující symbolickou konstantu, která bude optimalizována druhou fází evoluce, (viz. kapitola 3); f_{2s} , f_{2m} , a , a_1 , a_2 , $pair(a)$ jsou zástupné symboly terminálních

symbolů; $f_{2s} \in \Sigma^{2s}$ a $f_{2m} \in \Sigma^{2m}$ definují množiny terminálních symbolů, reprezentované dvouargumentovými funkcemi; $\Sigma^{2s} \in \{+, -\}$, $\Sigma^{2m} \in \{*, /\}$; $(a, a_1, a_2) \in \Sigma^{rv}$, pro které platí $a = a_1 = a_2$, $e \in \Sigma^{re}$. Obecně platí vztah $\Sigma^{rv} \subset \Sigma^{re} \subset \Sigma$; $\Sigma^{rv} = \{E_k, \dots, E_{k-5}, U_k, \dots, U_{k-5}, dE_k, \dots, dE_{k-4}, \}$ je množina terminálních symbolů *proměnných*, které jsou určeny k redukci; $\Sigma^{re} = \{+, -, *, /\} \cup \Sigma^{rv}$ je množina terminálních symbolů, které jsou určeny k redukci; $pair(a)$ je definován jako $f_{2m}(num, a) \mid f_{2m}(a, num)$.

Pravidla v uvedených tabulkách dovolují přepsat sekvenci terminálních symbolů na levé straně pravidla, jinou sekvencí terminálních symbolů, uvedených na pravé straně pravidla. Symboly uvedené v tabulkách, jsou významově totožné s gramatikou vyjádřenou v BNF [2].



P1 – P5: určují sadu pravidel podle které bude provedena redukce, viz. tabulka 8.

Obrázek 8: ART – vybrané typy možných redukcí

Princip metody spočívá v několikanásobném průchodu stromem a jeho postupné redukci, dle zadaného slovníku (*pravidel*). Některé z možných typů redukcí jsou ukázány na obrázku 8.

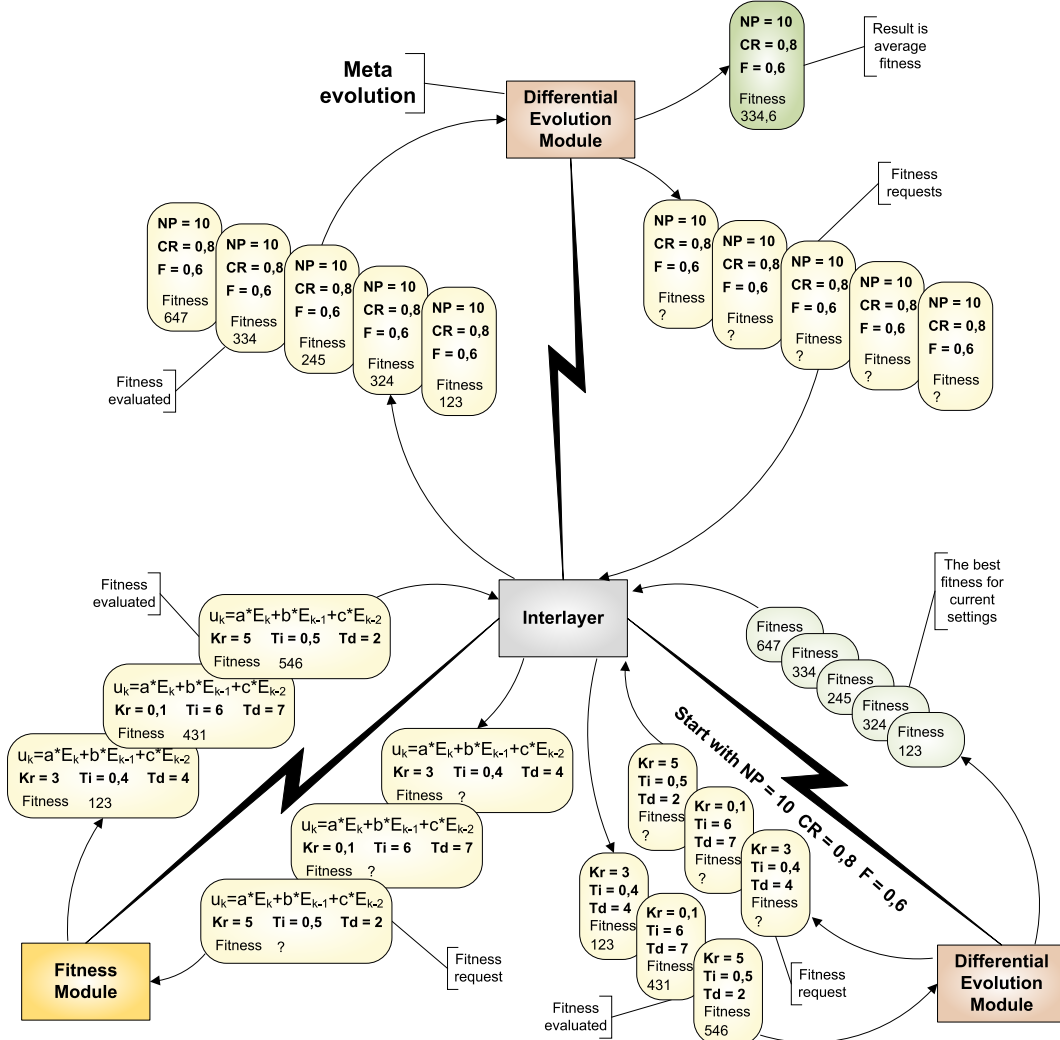
Na tomto obrázku jsou čtyři řešení ve formě stromu. Každý strom je očíslovaný. Výchozí stav stromu je na obrázku pod číslem 1, cílový stav je strom číslo 4. Redukce každé části jsou barevně odlišeny a kořenový uzel redukce je označen odkazem s číslem (P1 – P5). Tyto odkazy označují sadu pravidel, která byla použita k redukci, viz. tabulka 8.

Je vidět, že výchozí strom je složen z 21 uzlů a cílový ze 7 uzlů. Prakticky to neznamená jen kratší výsledné řešení, ale také přibližně trojnásobné *zrychlení výpočtu* během procházení stromu.

Metoda *ART* je volána po každém vytvoření jedince, během inicializace i během reprodukce, po volání metod křížení a mutace, viz. obrázek 7.

4 META-EVOLVE

Optimalizace symbolických konstant rovnice obecného regulátoru je velice zdlouhavý proces. Ve snaze o zefektivnění celého evolučního procesu byla vytvořena **meta-evolve**, která měla za cíl optimalizovat nastavení parametrů diferenciální evoluce z hlediska rychlosti a kvality nalezeného řešení. Cílem bylo nalézt optimální nastavení parametrů NP , CR , F .



Obrázek 9: Datový tok meta-evolve

Každý jedinec nadřazené vrstvy diferenciální evoluce obsahuje řešení „optimálního“ nastavení podřazené vrstvy diferenciální evoluce. Optimalizační proces pomocí evolučního algoritmu je heuristický proces a dopadne vždy s jiným výsledkem. Aby bylo možné jednotlivé řešení meta-evolve hodnotit, je nutné provést optimalizační proces s tímto nastavením. Pro zvýšení objektivitě ohodnocení řešení, je vhodné provést více hodnocení jednoho řešení. Tzn. pro jednoho jedince nadřazené vrstvy diferenciální evoluce spustit několikrát diferenciální evoluci podřazené vrstvy a výslednou hodnotu fitness určit například jako průměrnou hodnotu těchto ohodnocení.

Schéma datového toku meta-evolve je ukázáno na obrázku 9, kde je možno si všimnout dvou modulů diferenciální evoluce. Ve skutečnosti se jedná o dvě různé instance jednoho modulu, z nichž každá dělá něco jiného. První instance, označená *meta-evolve*, optimalizuje hodnoty parametrů druhé instance, která optimalizuje parametry PSD regulátoru. Popisy jednotlivých modulů jsou uvedeny v [32].

5 PRAKTICKÁ ČÁST

V této části jsou popsány konkrétní metody, které byly použity pro optimalizaci řídicích rovnic regulátorů. Je zde uveden také kompletní výpis pravidel gramatiky, které byly s různými variacemi použity pro generování struktury řídicích rovnic obecných regulátorů a také kompletní slovník sloužící pro snížení počtu redundantních částí ve struktuře řídicí rovnice obecného regulátoru.

5.1 Úplná gramatika

Gramatika představená v předchozích kapitolách byla v určitých ohledech pro jednoduchost ilustrativní. Finální a úplná G a G^{-1} gramatika použitá pro optimalizaci řídicích rovnic obecných regulátorů je uvedena v tabulkách 9, 10.

$\langle S \rangle$	$::=$	$\langle exp \rangle$
$\langle exp \rangle$	$::=$	$\langle fnc1 \rangle \langle exp \rangle \mid \langle fnc2 \rangle \langle exp \rangle \langle exp \rangle \mid \langle fnc2 \rangle \langle exp \rangle \langle pair \rangle \mid \langle pair \rangle \mid \langle var \rangle$
$\langle pair \rangle$	$::=$	$* num \langle var \rangle$
$\langle fnc1 \rangle$	$::=$	$U -$
$\langle mfnc1 \rangle$	$::=$	$U +$
$\langle fnc2 \rangle$	$::=$	$+ \mid - \mid * \mid /$
$\langle var \rangle$	$::=$	$U_{k-1} \mid U_{k-2} \mid U_{k-3} \mid U_{k-4} \mid U_{k-5} \mid$ $Y_k \mid Y_{k-1} \mid Y_{k-2} \mid Y_{k-3} \mid Y_{k-4} \mid Y_{k-5} \mid$ $E_k \mid E_{k-1} \mid E_{k-2} \mid E_{k-3} \mid E_{k-4} \mid E_{k-5} \mid$ $dE_k \mid dE_{k-1} \mid dE_{k-2} \mid dE_{k-3} \mid dE_{k-4}$

Tabulka 9: Pravidla generativní gramatiky pro optimalizaci regulátorů

NSM	$U_{k-1} \mid U_{k-2} \mid U_{k-3} \mid U_{k-4} \mid U_{k-5} ::= \langle var \rangle \mid num$
	$Y_k \mid Y_{k-1} \mid Y_{k-2} \mid Y_{k-3} \mid Y_{k-4} \mid Y_{k-5} ::= \langle var \rangle \mid num$
	$E_k \mid E_{k-1} \mid E_{k-2} \mid E_{k-3} \mid E_{k-4} \mid E_{k-5} ::= \langle var \rangle \mid num$
	$dE_k \mid dE_{k-1} \mid dE_{k-2} \mid dE_{k-3} \mid dE_{k-4} ::= \langle var \rangle \mid num$
	$num ::= \langle var \rangle$
	$U - ::= \langle mfnc1 \rangle$
	$U + ::= \langle fnc1 \rangle$
SSM	$+ \mid - \mid * \mid / ::= \langle fnc2 \rangle$
	$U + \mid U - ::= \langle var \rangle \mid \langle num \rangle$
GM	$+ \mid - \mid * \mid / ::= \langle var \rangle \mid \langle num \rangle \mid \langle fnc1 \rangle$
	$U_{k-1} \mid U_{k-2} \mid U_{k-3} \mid U_{k-4} \mid U_{k-5} ::= \langle exp \rangle$
	$E_k \mid E_{k-1} \mid E_{k-2} \mid E_{k-3} \mid E_{k-4} \mid E_{k-5} ::= \langle exp \rangle$
	$dE_k \mid dE_{k-1} \mid dE_{k-2} \mid dE_{k-3} \mid dE_{k-4} ::= \langle exp \rangle$
	$U + \mid U - ::= \langle exp \rangle$

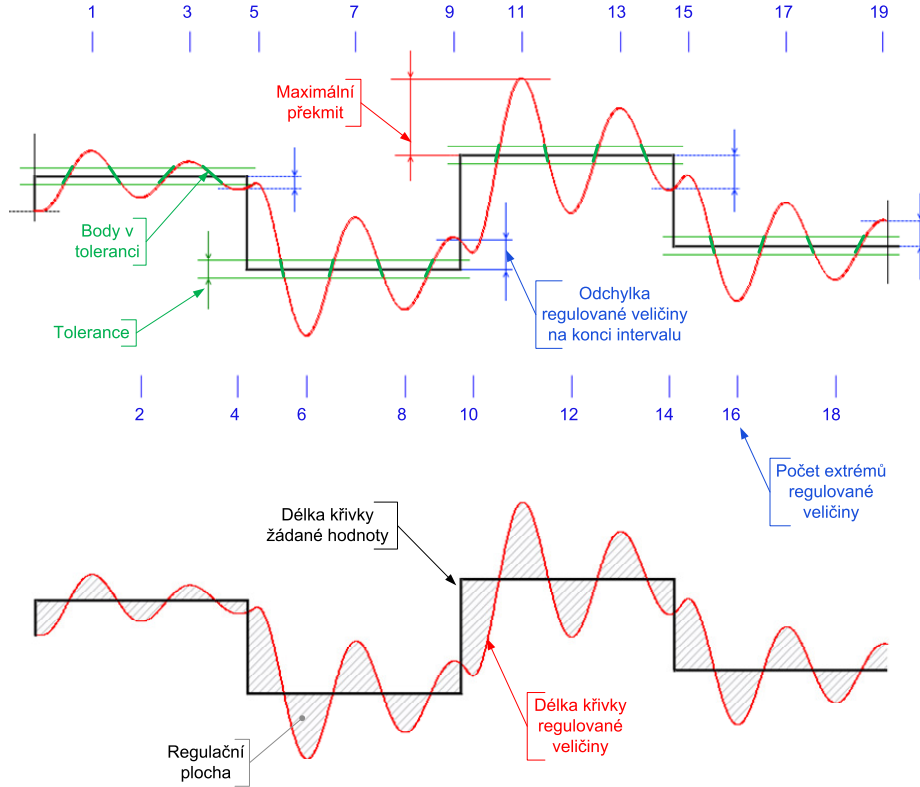
NSM: nestrukturální mutace (*Non-Structural Mutation*), **SSM:** Zkracující strukturální mutace (*Shortening Structural Mutation*), **GM:** obecná mutace (*General Mutation*)

Tabulka 10: Pravidla analytické gramatiky pro optimalizaci regulátorů

Z uvedených pravidel P , P^{-1} je patrné, že narozdíl od předchozích uvedených pravidel lze pomocí gramatiky přímo generovat pouze symbolické konstanty num , které jsou dále optimalizovány ve druhé fázi evoluce. Dále je vidět, že *terminální symbol* reprezentující *unární plus* ($U+$) nelze ze startovního symbolu vygenerovat. Terminál $U+$ lze vygenerovat pouze použitím operátoru mutace za pomoci neterminálu $\langle mfnc1 \rangle$. Z důvodu urychlení optimalizace přibyl také redundantní neterminál $\langle pair \rangle$. Také je možné si všimnout, že přibýly terminální symboly $Y_k, \dots, Y_{k-5}$, reprezentující hodnotu regulované veličiny v čase kT .

5.2 Kriteriační funkce

Vzhledem k optimalizaci řídicích rovnic regulátorů pomocí dvoufázové transplantační evoluce bylo nutné navrhnout kriteriační (ohodnocovací) funkci. Tento návrh nebyl vůbec snadnou záležitostí, jelikož optimalizace struktury regulátoru není tak přímočará jako optimalizace konstant PID (PSD) regulátorů.



Obrázek 10: Kriteria hodnocení

Problémem návrhu struktury řídicí rovnice regulátoru je nemožnost použít hodnocení kvality regulace v jednom intervalu, pro jednu hodnotu, $\dots$, jelikož v tomto případě by došlo k nalezení takového řešení, které sice dokáže nastavit řídicí veličinu na požadovanou velikost, ale to ještě neznamená, že daná struktura je regulátorem, resp. kvalitním regulátorem. Velice jednoduchým případem může být například rovnice regulátoru ve tvaru $u_k = 5$ ($u_k = \text{num}$; $\text{num} = 5$). V případě proporcionální regulované soustavy, kdy $k_s = 1$ a žádané hodnoty $w = 5$, sice časem dojde k nastavení regulované veličiny na hodnotu 5, ale tato struktura regulátoru je naprosto nepoužitelná. Tento příklad byl velice jednoduchou ukázkou toho, proč optimalizace obecných regulátorů není vůbec snadnou záležitostí. Pro hodnocení kvality regulace také nelze použít *pouze* integrálního kritéria, jelikož malá velikost regulační plochy ještě neznamená, že daná struktura regulátoru je kvalitní. Kvůli výše uvedenému je vhodné provádět simulaci regulace na relativně dlouhém intervalu, díky čemuž se prodlužuje doba optimalizace.

Pro ohodnocení průběhu regulace bylo použito multikriteriační funkce, která vyhodnocuje simulaci regulace z hlediska:

- velikost kvadratické regulační plochy: $I = \int_0^{t_s} e(t)dt$, kde $e(t)$ je regulační odchylka v čase t , t_s je délka simulace
- počet extrémů regulované veličiny
- průměrná chyba na konci jednotlivých intervalů

- počet bodů na křivce regulované veličiny v tolerančním pásmu
- poměr délek křivky regulované veličiny a řídicí veličiny (l_y/l_w)
- absolutní hodnota maximálního relativního překmitu regulované veličiny
- absolutní hodnota maximálního relativního překmitu akční veličiny vzhledem k povolenému intervalu akčního zásahu $\{u_{min}, u_{max}\}$

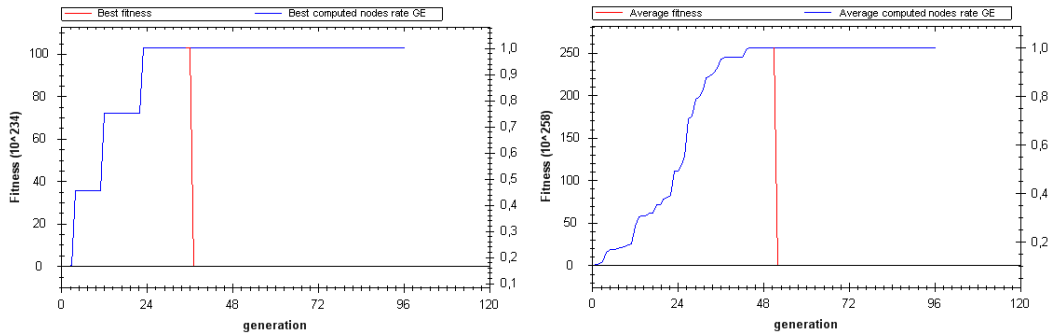
Uvedená kritéria jsou ukázána na obrázku 10. Snahou optimalizace je minimalizace regulační plochy, počtu extrémů regulované veličiny, průměrné relativní regulační odchylky na konci každého intervalu, relativního překmitu regulované, nebo akční veličiny. Dále je cílem optimalizování poměru délek křivek regulované a řídicí veličiny, tak aby $l_y/l_w = 1$ a maximalizace počtu bodů na křivce regulované veličiny, které jsou v povoleném tolerančním pásmu. Kromě těchto kritérií je ještě využíváno kritéria *ITAE*.

5.2.1 Hodnocení simulace v případě chyby

Aby bylo možno ohodnotit simulaci podle kritérií uvedených v kapitole 5.2, je nutné úspěšně dokončit celý proces simulace. V případě, že proces simulace nebude z jakéhokoliv důvodu dokončen (dělení nulou, $\pm\infty$, apod.), musí být algoritmus schopen hodnotit simulaci podle jiných kritérií. Tyto kritéria se liší v závislosti na optimalizovaném problému.

V případě, že je spuštěna **optimalizace PSD regulátoru**, je fitness hodnoceno podle počtu vypočítaných cyklů rekurentní rovnice PSD regulátoru. Přesněji podle poměru (c_c/c_{cp}) , kde c_c je počet skutečně vypočítaných cyklů a c_{cp} je předpokládaný počet cyklů.

Příklad: doba simulace je 10s, vzorkování 0,1s. Z toho vyplývá, že celkový předpokládaný počet výpočtů rekurentní rovnice PSD regulátoru je 100. V případě, že simulace bude z jakéhokoliv důvodu přerušena během výpočtu rekurentní rovnice v čase 8 sekund, bude poměr $c_c/c_p = 79/100 = 0,79$ (79 úspěšně dokončených výpočtů).



Obrázek 11: Průběh optimalizace PSD regulátoru pomocí diferenciální evoluce

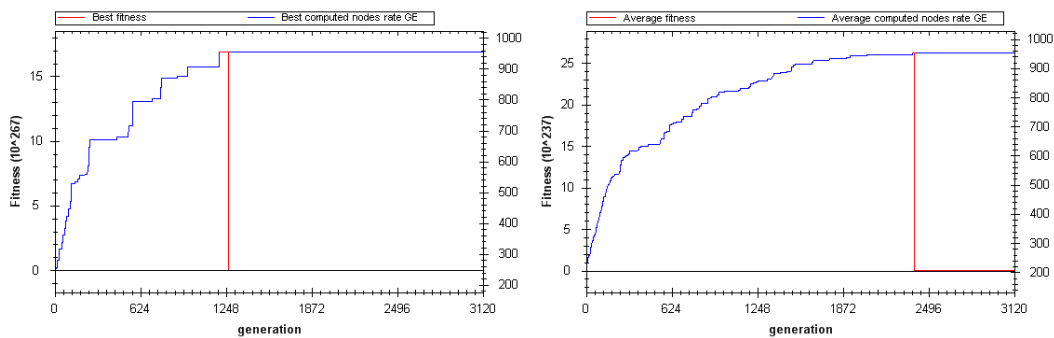
Optimalizace PSD regulátoru pomocí diferenciální evoluce je ukázána na obrázku 11. Na levé straně je graf vývoje nejlepšího řešení a na pravé straně je zobrazen vývoj průměrného fitness v populaci. Na obou grafech je možno vidět dvě křivky. *Modrá* křivka zobrazuje poměr c_c/c_{cp} . *Červená* křivka zobrazuje vypočítané fitness v případě, že se podařilo úspěšně dokončit celou simulaci regulace. Je vidět, že fitness bylo možné vyčíslit až v případě, kdy $c_c/c_{cp} = 1$.

Jestliže je spuštěna **optimalizace obecného regulátoru**, je fitness počítáno podle počtu propočítaných uzlů ve stromu jedince za dobu simulace regulačního pochodu. Přesněji podle poměru (c_n/c_{nc}) , kde c_n je skutečně propočítaný počet uzlů a c_{nc} je počet uzlů ve stromu jedince.

Příklad: Doba simulace je 10s, vzorkování 0,1s, rovnice regulátoru je ve tvaru $u_k = 3 \cdot E_k$. Z uvedených rovnic vyplývá, že strom funkce má 3 uzly ($c_{nc} = 3$), (viz. kapitola 2). Při každém úspěšném výpočtu rekurentní rovnice tedy dojde k průchodu a propočtu 3 uzlů. V případě, že simulace bude

z jakéhokoliv důvodu přerušena během výpočtu rekurentní rovnice v čase 8 sekund, při výpočtu součinu uzlů 3 a E_k bude výsledný počet propočítaných uzlů $79 \cdot 3 + 1 \cdot 2$ (79 krát úspěšně dokončen výpočet 3 uzlů a v posledním výpočtu byly vyčísleny dva uzly a u výpočtu jejich součinu došlo k chybě). Z toho vyplývá, že poměr $c_n/c_{nc} = 239/3 = 79,6\bar{6}$. V případě, že tento poměr vydělíme délkou simulace dělenou vzorkovací periodou, dostaneme poměr stejný jako v případě optimalizace PSD regulátoru ($79,66/100 = 0,7966$). Počítání uzlů namísto počtu úspěšně vypočítaných cyklů rekurentní rovnice je přesnější. Toho si lze všimnout i porovnáním s předchozím příkladem.

Optimalizace obecného regulátoru pomocí dvoufázové transplantační evoluce je ukázána na obrázku 12. Na levé straně je graf vývoje nejlepšího řešení a na pravé straně je zobrazen vývoj průměrného fitness v populaci. Na obou grafech je možno vidět dvě křivky. *Modrá* křivka zobrazuje poměr c_n/c_{nc} . *Červená* křivka zobrazuje vypočítané fitness v případě, že se podařilo úspěšně dokončit celou simulaci regulace. Z obrázku je mimo jiné vidět, že fitness bylo možné vyčíslit až v případě, kdy $c_n/c_{nc} = l_s \cdot c_{nc}$, kde l_s je délka simulace.



Obrázek 12: Průběh optimalizace parametrů obecného regulátoru metodou diferenciální evoluce

To, že je **simulace úspěšně dokončena**, ještě neznamená, že se podaří vypočítat fitness. Vzhledem k tomu, že fitness funkce provádí multikriteriální hodnocení simulace, může se stát, že v nějaké části výpočtu dojde k dosažení $\pm\infty$. V tom případě, je fitness počítáno podle *počtu úspěšně vypočítaných kritérií*.

Z výše uvedeného vyplývá, že ohodnocení simulace regulace může probíhat třemi způsoby, které jsou uvedeny níže v pořadí od nejhoršího případu, k nejlepšímu:

1. simulaci se nepodařilo z nějakého důvodu dokončit a fitness je počítáno z poměru c_c/c_{cp} , nebo c_n/c_{nc}
2. simulace byla úspěšně dokončena, ale nepodařilo se úspěšně vyčíslit všechny kritéria fitness funkce. Fitness je počítáno podle počtu úspěšně vyhodnocených kritérií.
3. simulace i výpočet fitness se podařilo úspěšně dokončit

Z uvedeného výčtu vyplývá, že *jedinec 3* má vždy lepší fitness, než *jedinec 2* a ten má lepší fitness, než *jedinec 1*, ($f_{i3} < f_{i2} < f_{i1}$). Hodnocení podle počtu úspěšně vypočítaných kritérií fitness funkce není v grafech uvedeno.

Z obrázků 11 a 12 je vidět, že tento typ hodnocení je velice důležitý a bez něj by se s vysokou pravděpodobností vůbec nepodařilo strukturu obecného regulátoru, případně parametry PSD regulátoru optimalizovat, jelikož by jedince nebylo podle čeho hodnotit. Zvláště důležitý je tento typ hodnocení během optimalizace obecného regulátoru, viz. obrázek 12. Z tohoto obrázku je vidět, že celý průběh simulace pro jednoho jedince dvoufázové transplantační evoluce se podařilo dokončit až po cca 1200 generacích diferenciální evoluce a průměrné fitness v populaci se podařilo dle kritériální funkce hodnocící průběh simulace vyčíslit dokonce až po 2400 generacích.

5.3 Modifikované schéma DE

V kapitole 3 bylo řečeno, že k optimalizaci symbolických konstant ve struktuře jedince dvoufázové transplační evoluce byla použita diferenciální evoluce. Tento typ evolučního algoritmu byl na počátku vývoje testován se schémata *DE/Rand/1* a *DE/Best/1* [32]. V průběhu testování nastal případ, kdy tyto schémata nepodávaly očekávané výsledky, hlavně v případech optimalizace parametrů z intervalu $\mathbb{R}$. Ve snaze vyřešit tento problém vznikla modifikace těchto schémat. Nové schémata byly pojmenovány **DE/Rand/1_mod** a **DE/Best/1_mod**, případně **DE/Best/1plus**, **DE/Rand/1plus** [36].

Rozdíl oproti schématům *DE/Rand/1* a *DE/Best/1* je ve způsobu výpočtu **diferenčního vektoru**. *Šumový vektor* je definován ve tvaru:

$$\vec{v}_i(t+1) = \vec{x}_{r_1^i}(t) + \beta \cdot f(\vec{x}_{r_2^i}(t), \vec{x}_{r_3^i}(t)) \quad (5.1)$$

Diferenční vektor je v tomto případě funkcí:

$$f(\vec{x}_{r_2^i}(t), \vec{x}_{r_3^i}(t)) \quad (5.2)$$

Výpočet tohoto vektoru závisí na hodnotách jednotlivých dimenzí vektorů $\vec{x}_{r_2^i}(t)$ a $\vec{x}_{r_3^i}(t)$:

$$f(\vec{x}_{r_2^i}, \vec{x}_{r_3^i}) = \begin{cases} \vec{x}_{r_2^i} - \vec{x}_{r_3^i} & \text{pro } [(\vec{x}_{r_2^i} \geq 0) \wedge (\vec{x}_{r_3^i} \geq 0)] \vee [(\vec{x}_{r_2^i} \leq 0) \wedge (\vec{x}_{r_3^i} \leq 0)] \\ \vec{x}_{r_2^i} + \vec{x}_{r_3^i} & \text{jinak} \end{cases} \quad (5.3)$$

Zápis byl pro přehlednost zjednodušen a platí $f(\vec{x}_{r_2^i}, \vec{x}_{r_3^i}) \Leftrightarrow f(\vec{x}_{r_2^i}(t), \vec{x}_{r_3^i}(t))$. Z rovnice 5.3 je patrné, že výpočet rozdílového vektoru závisí znaménku jednotlivých dimenzí rozdílových vektorů.

S odstupem času, navíc po optimalizaci parametrů diferenciální evoluce, je přínos těchto modifikací diskutabilní, jelikož po vytvoření *meta-evoluce*, která slouží pro optimalizaci parametrů diferenciální evoluce, dávají klasická schémata podobné výsledky, jako jejich modifikace. Více o optimalizaci parametrů diferenciální evoluce diskutuje kapitola 4.

6 EXPERIMENTÁLNÍ ČÁST

V následujících kapitolách jsou prezentovány ukázky regulačních pochodů, včetně jejich vzájemného porovnání. V úvodu každé podkapitoly je vždy uvedena soustava, pro kterou proběhla simulace a je popsáno co na této soustavě bylo pozorováno. Porovnávány jsou různé možnosti seřízení regulátorů, z nichž je větší váha kladena na evoluční metody. Nejdůležitější z této části jsou kapitoly pojednávající o optimalizaci struktury, včetně parametrů obecných regulátorů, (viz. kapitola 6.3). Algebraické metody seřízení regulátorů jsou uvedeny například v [3, 31].

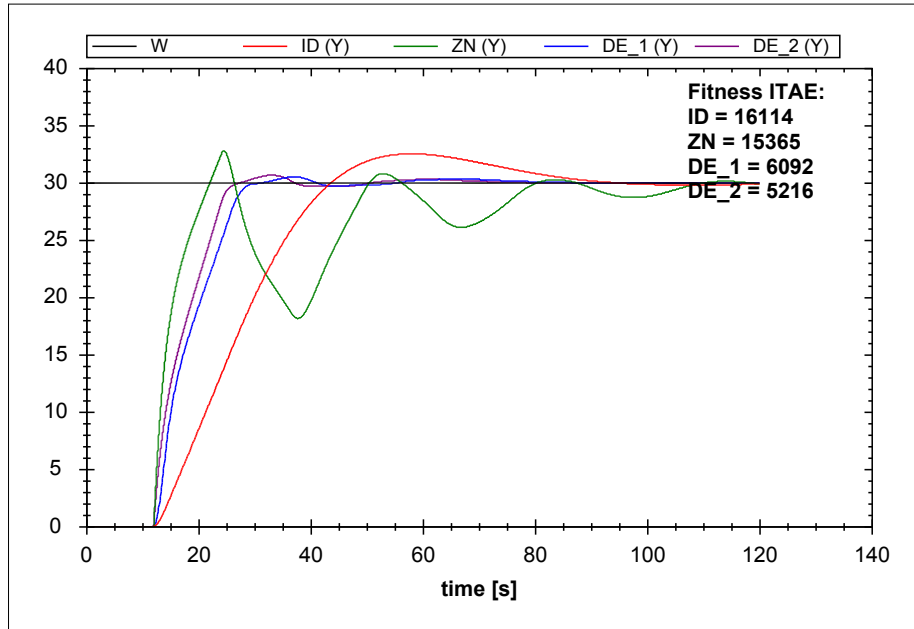
6.1 Porovnání regulačních pochodů PSD regulátorů pro různé metody nastavení parametrů

Pro proporcionální soustavu danou přenosem (Eq. 6.1)

$$G_S(S) = \frac{2}{(5s + 1)(2s + 1)} e^{-12s} \quad (6.1)$$

bylo provedeno porovnání regulačních pochodů *neupravených* PSD regulátorů, jejichž parametry byly nastaveny metodou:

1. inverze dynamiky (metoda požadovaného modelu), pro $T = 2s$
2. Ziegler-Nichols (metoda kritického stavu), pro $T = 0,1s$
3. diferenciální evoluce, pro $T = 2s$ a včetně T



ID: parametry PSD nastaveny metodou *inverze dynamiky* pro $T = 2s$, **ZN:** parametry PSD nastaveny metodou *Ziegler-Nicholse* pro $T = 0,1s$, **DE_1:** parametry PSD nastaveny pomocí *diferenciální evoluce* pro $T = 2s$, **DE_2:** parametry PSD nastaveny pomocí *diferenciální evoluce*, včetně T ($T = 0,02s$)

Obrázek 13: Porovnání regulačních pochodů PSD regulátorů

Kriteriální funkce hodnotila kvalitu regulačního pochodu dle kritérií uvedených v kapitole 5.2.

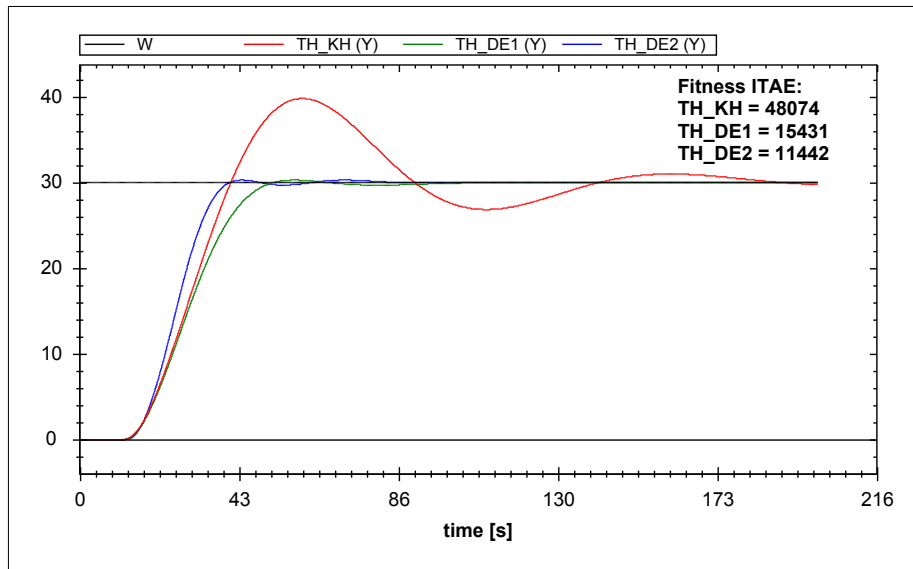
6.2 Porovnání regulačních pochodů Takahashiho regulátorů pro různé metody nastavení parametrů

Pro proporcionální soustavu z kapitoly 6.1 danou přenosem (Eq. 6.1)

$$G_S(S) = \frac{2}{(5s + 1)(2s + 1)} e^{-12s} \quad (6.2)$$

bylo provedeno porovnání regulačních pochodů *Takahashiho regulátorů*, jejichž parametry byly nastaveny:

1. z kritických hodnot přechodové charakteristiky soustavy, pro $T = 2s$
2. diferenciální evolucí, pro $T = 2s$ a $T = 0,1s$



TH_KH: parametry Takahashiho regulátoru nastaveny z kritických hodnot pro $T = 2s$, **TH_DE1:** parametry Takahashiho regulátoru nastaveny z kritických hodnot pro $T = 2s$, **TH_DE2:** parametry Takahashiho regulátoru nastaveny z kritických hodnot pro $T = 0,1s$

Obrázek 14: Porovnání regulačních pochodů Takahashiho regulátoru

6.3 Optimalizace obecných regulátorů

V této kapitole budou ukázány příklady optimalizace obecných regulátorů metodou *TLTE* s různými typy pravidel gramatiky a kritériálními funkcemi. Regulační pochody budou demonstrovány na proporcionální soustavě se setrvačností druhého řádu, jejíž přenos je:

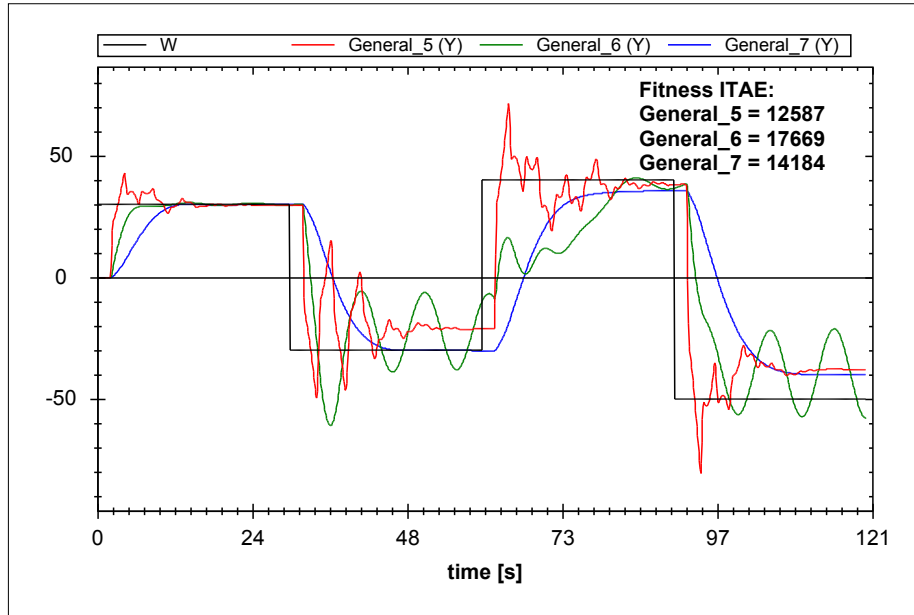
$$G_S(S) = \frac{4}{(3s + 1)(5s + 1)} e^{-2s} \quad (6.3)$$

Dvoufázová transplantační evoluce: Pro optimalizaci struktury a parametrů řídicí rovnice *obecných regulátorů* byla použita metoda *TLTE*. Parametry *TE* byly nastaveny: Počet jedinců = 50, počet nově vytvořených jedinců v každé generaci = 25, maximální počet uzlů jedince = 30, pravděpodobnost křížení = 90%, pravděpodobnost mutace rodičů = 25%, pravděpodobnost mutace potomků = 25%.

Pro optimalizaci parametrů byla použita diferenciální evoluce s parametry $G = 1500$, $NP = 9$, $F = 1,07$, $CR = 0,58$, schéma: DE/Rand/1_mod.

6.3.1 Ukázky chybně nastavené optimalizace

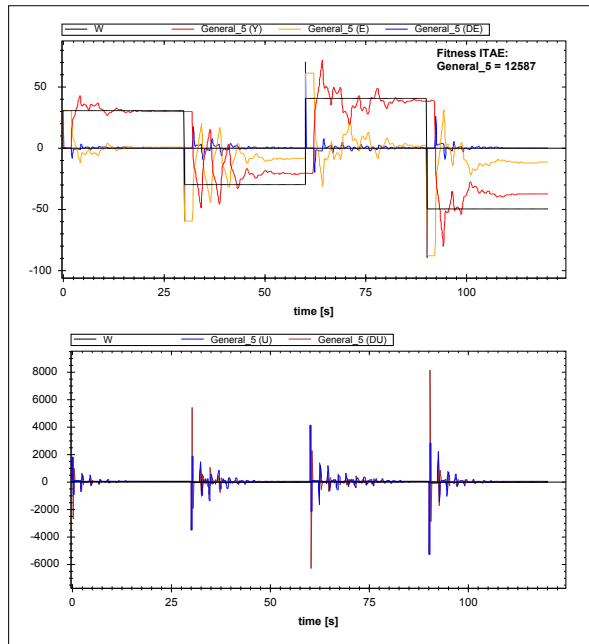
Na obrázku 15 jsou ukázky regulačních pochodů obecných regulátorů, jenž jsou výsledkem chybně nastavené optimalizace obecných regulátorů. V průběhu mnoha experimentů vyšlo najevo pár podstatných faktů o nastavení optimalizačního procesu. Nejdůležitějšími postřehy jsou:



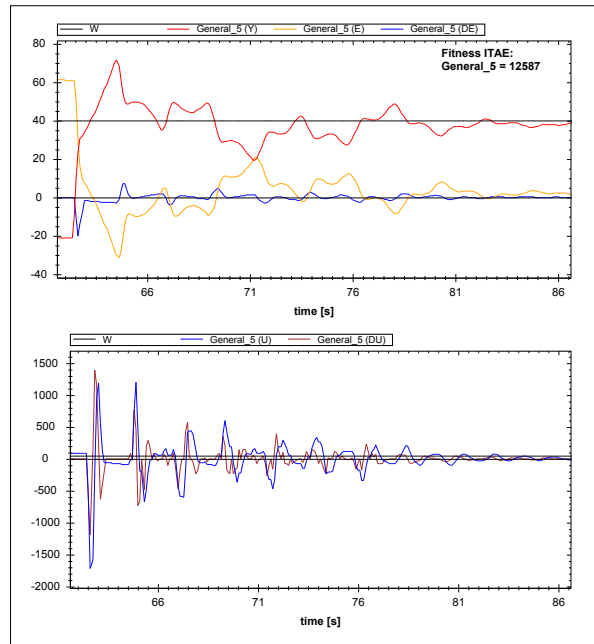
Obrázek 15: Regulační pochody chybně optimalizovaných obecných regulátorů

1. V případě použití kritéria IAE [32] je důležité současně hodnotit a penalizovat i kmitání akčního zásahu, případně regulované veličiny. Důkazem je regulační pochod regulátoru označeného *General_5*, viz. obrázek 16, který byl během optimalizace hodnocen pouze kritériem IAE a optimalizován pouze pro jeden interval s žádanou hodnotou $w = 30$.
2. Předcházející podmínka je podmínkou nutnou, nikoliv postačující. Tímto důkazem je regulační pochod regulátoru *General_6*. Tento regulátor byl během optimalizace hodnocen kritériem IAE a bylo penalizováno kmitání akčního zásahu. Optimalizace proběhla na jednom intervalu pro jednu žádanou hodnotu $w = 30$. Z obrázku 15, 17 je vidět, že nastavení regulované veličiny na žádanou hodnotu $w = 30$ proběhlo bez problémů, ale pro nastavení jiné žádané hodnoty je regulátor naprosto nevyhovující. Z tohoto vyplývá, že nestačí provést optimalizaci pro jednu hodnotu žádané veličiny.
3. Křivka *General_7*, na obrázku 15, 18 ukazuje regulační pochod regulátoru který byl během optimalizace hodnocen kritériem IAE , bylo penalizováno kmitání akčního zásahu, průměrná regulační odchylka na konci každého intervalu, počet bodů v toleranci a délka křivky, (viz. kapitola 5.2). Regulátor byl optimalizován pro dva intervaly, z nichž jeden byl $w = 30$ a druhý byl $w = -30$. Z obrázku je patrné, že regulátor dokáže bez problémů nastavit regulovanou veličinu na hodnotu $y = \pm 30$, ale pro jakoukoliv jinou žádanou hodnotu w je nepoužitelný. Z toho vyplývá, že nestačí optimalizovat regulátor na dvou intervalech, ve kterých je žádaná hodnota regulované veličiny $w = \pm a$, kde $w_1 = a$ je žádaná hodnota regulované veličiny v prvním intervalu a $w_2 = -a$ je žádaná hodnota regulované veličiny ve druhém intervalu.

General_5: Regulační pochod *General_5* přísluší regulátoru s řídicí rovnicí. Tato rovnice je výsledkem optimalizace po 75. generaci:



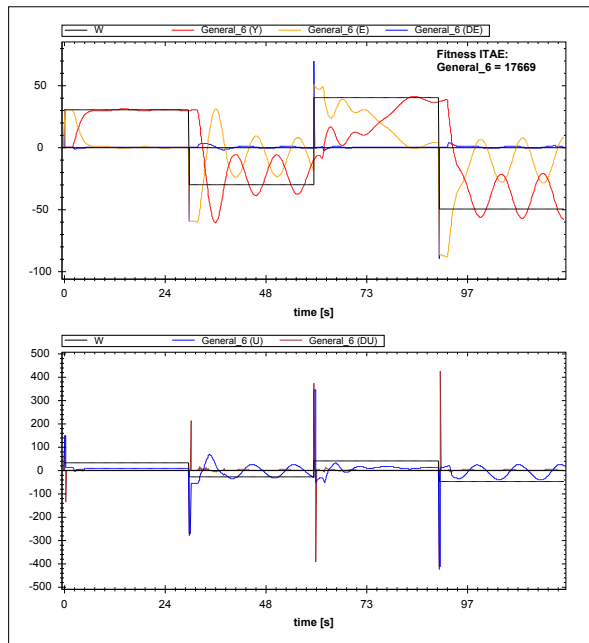
(a) Celý regulační pochod



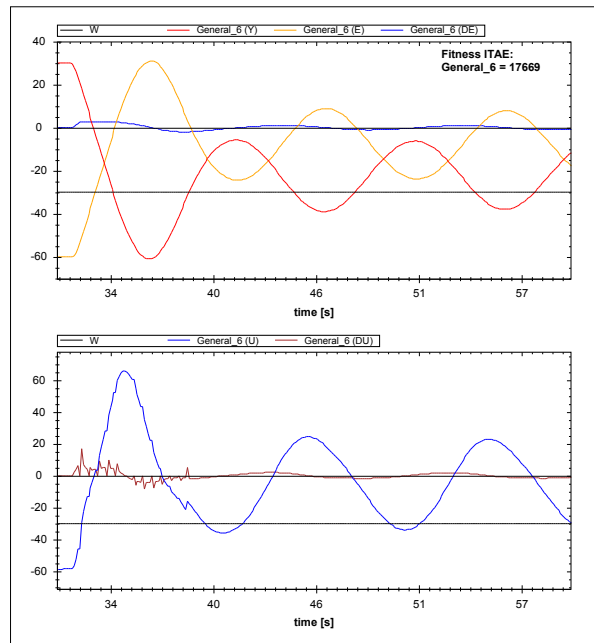
(b) Detail regulačního pochodu

Horní graf: průběh regulačního pochodu, **Dolní graf:** průběh akčních zásahů, **červená (Y):** regulovaná veličina, **oranžová (E):** regulační odchylka, **modrá (DE):** derivace regulační odchylky, **modrá (U):** akční zásah regulátoru, **hnědá (DU):** derivace akčního zásahu, **černá (W):** žádaná hodnota regulované veličiny

Obrázek 16: Regulační pochod obecného regulátoru *General_5*



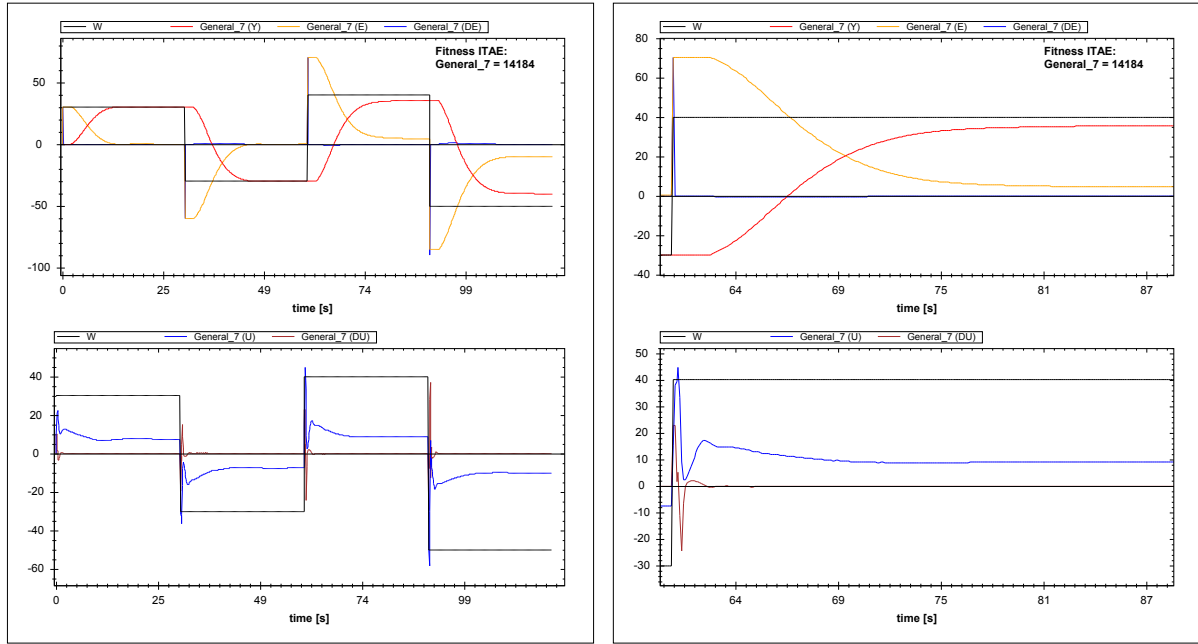
(a) Celý regulační pochod



(b) Detail regulačního pochodu

Horní graf: průběh regulačního pochodu, **Dolní graf:** průběh akčních zásahů, **červená (Y):** regulovaná veličina, **oranžová (E):** regulační odchylka, **modrá (DE):** derivace regulační odchylky, **modrá (U):** akční zásah regulátoru, **hnědá (DU):** derivace akčního zásahu, **černá (W):** žádaná hodnota regulované veličiny

Obrázek 17: Regulační pochod obecného regulátoru *General_6*



(a) Celý regulační pochod

(b) Detail regulačního pochodu

Horní graf: průběh regulačního pochodu, **Dolní graf:** průběh akčních zásahů, **červená (Y):** regulovaná veličina, **oranžová (E):** regulační odchylka, **modrá (DE):** derivace regulační odchylky, **modrá (U):** akční zásah regulátoru, **hnědá (DU):** derivace akčního zásahu, **černá (W):** žádaná hodnota regulované veličiny

Obrázek 18: Regulační pochod obecného regulátoru *General_7*

$$U_k = (((32,5582238063037 * E_{k-5}) + (58,7985461639848 * E_k)) + (7,08170165073634 + (-89,9739865620223 * E_{k-2})))$$

Optimalizace struktury proběhla s pravidly definovanými v kapitole (viz. kapitola 5.1), ze kterých byly vyloučeny terminální symboly $Y_k, \dots, Y_{k-1}$ a „/“.

Regulační pochod, včetně zobrazeného detailu v intervalu od $\approx 62s$ do $\approx 68s$ je na obrázku 16a, resp. 16b.

General_6: Regulační pochod *General_6* přísluší regulátoru s řídicí rovnicí. Tato rovnice je výsledkem optimalizace po 121. generaci:

$$U_k = (7,62193225390625 + ((E_k * 4,66547982026174) + (E_{k-4} * (-3,41207562721883 + (((-((45,7536105413915 + E_k) + 8,54175596253857)) * (E_k * 0,0305737222361965) * 0,0148294215892667))))))$$

Optimalizace struktury proběhla s pravidly definovanými v kapitole 5.1 ze kterých byly vyloučeny terminální symboly $Y_k, \dots, Y_{k-1}$ a „/“.

Regulační pochod, včetně zobrazeného detailu v intervalu od $\approx 32s$ do $\approx 59s$ je na obrázku 17a, resp. 17b.

General_7: Regulační pochod *General_7* přísluší regulátoru s řídicí rovnicí:

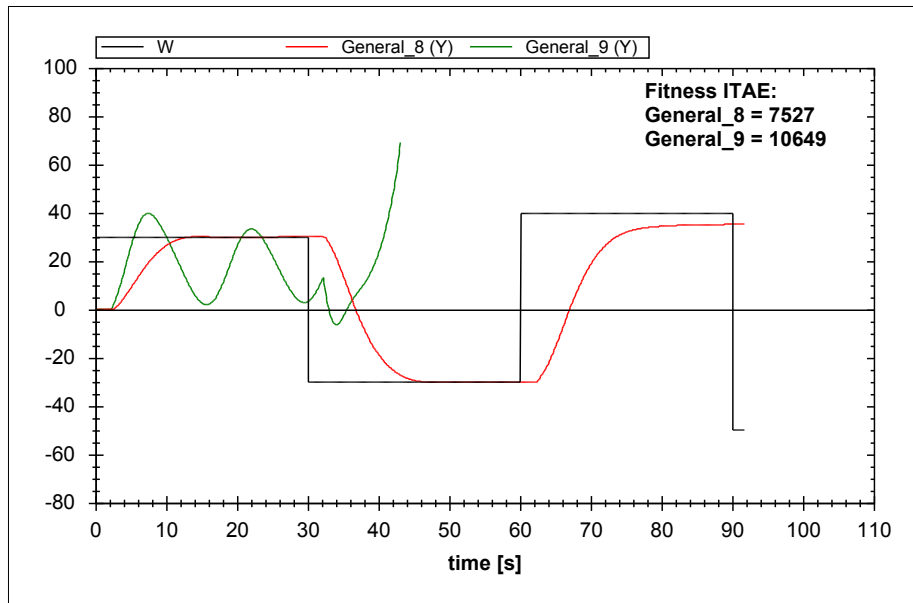
$$U_k = (((-((-(-E_{k-2} * (-0,301663064996273))) - (U_{k-1} * 1,03071349951939))) + ((U_{k-1} * (-7,23369179910225E-05)) * ((-E_k * (((-((-DE_{k-3} / (-37,8122868930685))) - U_{k-1})) / (-31,2077417822251)))) - (U_{k-3} * (U_{k-4} * (-7,4022042108816)))))) - (E_k * (-0,325440780327943)))$$

Optimalizace struktury proběhla s pravidly definovanými v kapitole 5.1, ze kterých byly vyloučeny terminální symboly $Y_k, \dots, Y_{k-1}$.

Regulační pochod, včetně zobrazeného detailu v intervalu od $\approx 61s$ do $\approx 88s$ je na obrázku 18a, resp. 18b.

6.3.2 Ukázky chyb v průběhu simulace

Na obrázku 19 jsou ukázány dva příklady chyb v průběhu simulace regulace. V obou zobrazených případech pokračuje regulovaná, příp. akční veličina do nekonečna. V případě *General_8* dosáhne $-\infty$ v čase 57,5s poté co se regulátor snaží vyrovnat prudce narůstající chybu regulované veličiny. Regulovaná veličina regulátoru *General_9* dosáhne hodnoty ∞ v 90,7s díky náhlému zákmitu akční veličiny. Z tohoto obrázku mj. vyplývá důležitost použití několikafázového ohodnocení regulačního pochodu, které se uplatňuje právě v případě takovýchto chyb, (viz. kapitola 5.2.1).



červená: General_8, zelená: General_9

Obrázek 19: Ukázky chyb ($\pm\infty$) v průběhu simulace regulace

General_8: Tento regulační pochod je výsledkem řídicí rovnice regulátoru:

$$U_k = ((((-(\text{DEk}_3 * (-0,554605924578443))) - (U_{k-1} * (-1,03526653482844))) + ((U_{k-1} * (U_{k-1} * 0,000491656195958113)) * ((- (U_{k-4})) - (U_{k-3} * 0,25300123214001)))) - (E_k * (-0,0236843739347194)))$$

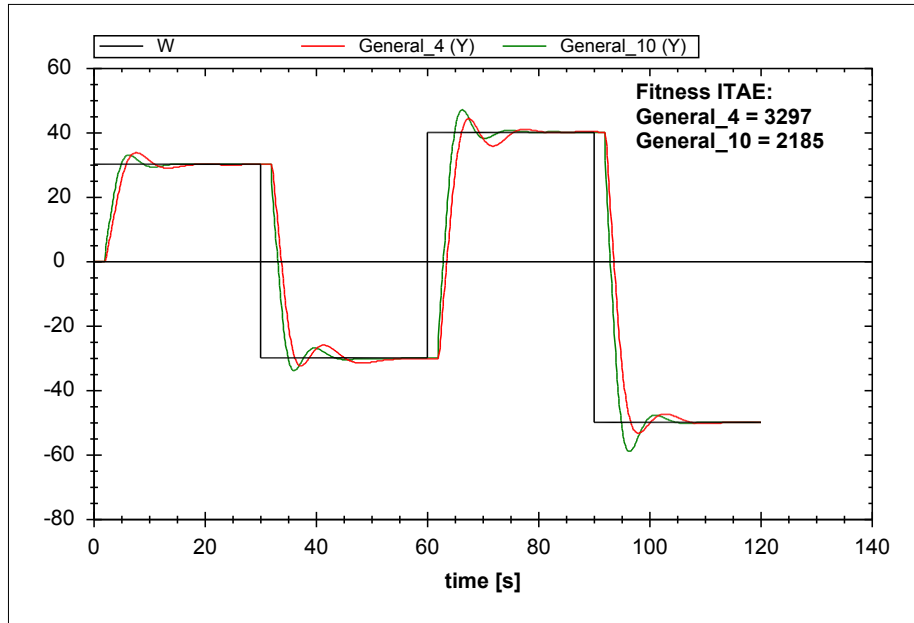
General_9: Regulační pochod je výsledkem řídicí rovnice regulátoru:

$$U_k = (-((((((-47,0642949272105) + (\text{DEk}_1 * \text{DEk}_4)) * (-3,50176842352982)) + ((- (E_k * 2 * (E_k / 1,41946999419199))) - (E_k * 2 * 6,04369731616662)))) * 0,0455554709203227) + ((E_k * 2 * ((U_{k-1} + (\text{DEk}_2 * (-76,3179821444081))) - (E_k * 79,2532577811589)) * (-0,000180769843021228))) - ((E_k * 2 * (\text{DEk}_2 * (-0,00188164990393764))) * (\text{DEk}_2 * (-4,62774508630033)))))$$

Tento regulátor je opět jedním z příkladů, který dokazuje, že nestačí optimalizovat regulátor na dvou intervalech, ve kterých je žádaná hodnota regulované veličiny $w = \pm a$, kde $w_1 = a$ je žádaná

hodnota regulované veličiny v prvním intervalu a $w_2 = -a$ je žádaná hodnota regulované veličiny ve druhém intervalu. Z obrázku 19 je patrné, že pro první dva intervaly, ve kterých byla žádaná hodnota $w = 30$, resp. $w = -30$ proběhla regulace bez jakýchkoliv problémů.

6.3.3 Porovnání kritériálních funkcí



červená (**General_4**): Kritéria hodnocení 5.2, zelená (**General_10**): Kritérium *ITAE*

Obrázek 20: Porovnání kritériálních funkcí

V této kapitole jsou srovnány různé možnosti hodnocení regulačních pochodů. V případě, že se použijí kritéria hodnocení dle kapitoly 5.1, je možno dosáhnout větší flexibility nastavení regulátoru. Na druhou stranu se jedná o mnohdy náročnou implementaci mnoha kritérií a neméně důležité sestavení samotné kritériální funkce. Z tohoto důvodu je výhodnější použít kritérium *ITAE*, příp. *ITSE* a k němu v případě potřeby přidat jiné doplňující kritéria.

Na obrázku 20 jsou vidět regulační pochody regulátoru *General_4* a *General_10*.

General_4: je regulační pochod obecného regulátoru s řídicí rovnicí:

$$U_k = (U_{k-1} - ((2,66414343914243 * E_{k-4}) + (((DE_k * (-3,28788061116549)) + (((((DE_k * 76,1825131087609) + 1,26443835876666) - ((-1,27123447444844) * E_{k-1})) * ((9,32622370528897E-07) * E_{k-5})) * E_{k-1})) - (2,6711609052335 * E_{k-5}))))$$

Tato rovnice je výsledkem optimalizačního procesu s kritériální funkcí hodnotící podle (viz. kapitola 5.2).

General_10: je regulační pochod obecného regulátoru s řídicí rovnicí:

$$U_k = (U_{k-1} + ((37,7949275202727 * E_k) + ((((-85,7168395987696) * E_{k-1}) + (51,5425586964905 * E_{k-2})) + (E_{k-5} * (-3,61245065902024)))))$$

Tato rovnice je výsledkem optimalizačního procesu s kritériální funkcí hodnotící podle *ITAE*.

Z obrázku je vidět, že pomocí kritérií z kapitoly 5.2 lze dosáhnout obdobného výsledku jako v případě použití kritéria *ITAE*.

6.4 Porovnání obecných regulátorů se setrvačností prvního řádu a PSD regulátorů – kmitavá proporcionální soustava bez dopravního zpoždění

Pro proporcionální soustavu s kmitavým průběhem, bez dopravního zpoždění, danou přenosem:

$$G_S(S) = \frac{k_S}{T_0^2 s^2 + 2\xi T_0 s + 1} \quad (6.4)$$

s parametry $k_S = 0.5$, $T_0 = 1$, $\xi = 0.1$ bylo provedeno porovnání regulačních pochodů *PSD regulátoru* a *obecného regulátoru*. Parametry *PSD regulátoru* byly optimalizovány *diferenciální evolucí*, struktura a parametry obecných regulátorů byly optimalizovány *TLTE* (viz. kapitola 3). Vzorkovací frekvence byla v případě regulátoru *General_12* zvolena $T = 0.1s$. Regulátory *General_13* a *PSD_2_DE* měly vzorkovací periodu nastavenou na $T = 0.01s$. Limit akčního zásahu byl u všech regulátorů nastaven v rozsahu ± 200 . Pásmo necitlivosti regulátoru nastaveno na 1% žádané hodnoty, při zpětné diferencii regulované veličiny $\nabla y(kT) = 0.1$. Během regulace byl také použit filtr druhého řádu.

Pro optimalizaci struktury a parametrů obecných regulátorů byla využita sada pravidel definovaná v kapitole 5.1. Z této sady byly v případě optimalizace regulátoru *General_12* odebrány pravidla obsahující terminální symboly „/“, $Y_{k-3}, \dots, Y_{k-5}$, $E_{k-3}, \dots, E_{k-5}$, $dE_k, \dots, dE_{k-4}$ a pravidla obsahující $U_{k-1}, \dots, U_{k-5}$. V případě optimalizace regulátoru *General_13* byly odebrány pravidla „/“, $dE_k, \dots, dE_{k-4}$ a pravidla obsahující $U_{k-1}, \dots, U_{k-5}$.

Diferenciální evoluce: Pro optimalizaci parametrů *PSD regulátoru* (k_R , T_I , T_D) byla použita *DE* s nastavením $G = 1500$, $NP = 20$, $F = 1.07$, $CR = 0.58$, schéma: *DE/Rand/1*. Výsledkem optimalizace je nastavení parametrů *PSD regulátoru* $k_R = 3858.862378$, $T_I = 0.510784$, $T_D = 0.122012 \cdot 10^{-2}$.

Dvoufázová transplantační evoluce: Pro optimalizaci struktury a parametrů řídicí rovnice *obecného regulátoru* byla použita metoda *TLTE*. Parametry *TE* byly nastaveny: Počet jedinců = 30, počet nově vytvořených jedinců v každé generaci = 15, maximální počet uzlů jedince = 30, pravděpodobnost křížení = 90%, pravděpodobnost mutace rodičů = 25%, pravděpodobnost mutace potomků = 25%, metoda *ART*, (viz. kapitola 3.1), byla aktivní. Parametry *MDE* byly nastaveny: $G = 1500$, $NP = 20$, $F = 1.07$, $CR = 0.58$, schéma: *DE/Rand/1* (pro *General_12*), *DE/Rand/1_mod* (pro *General_13*). Výsledkem optimalizace v 7. generaci *TLTE* je řídicí rovnice *obecného regulátoru General_12* ve tvaru:

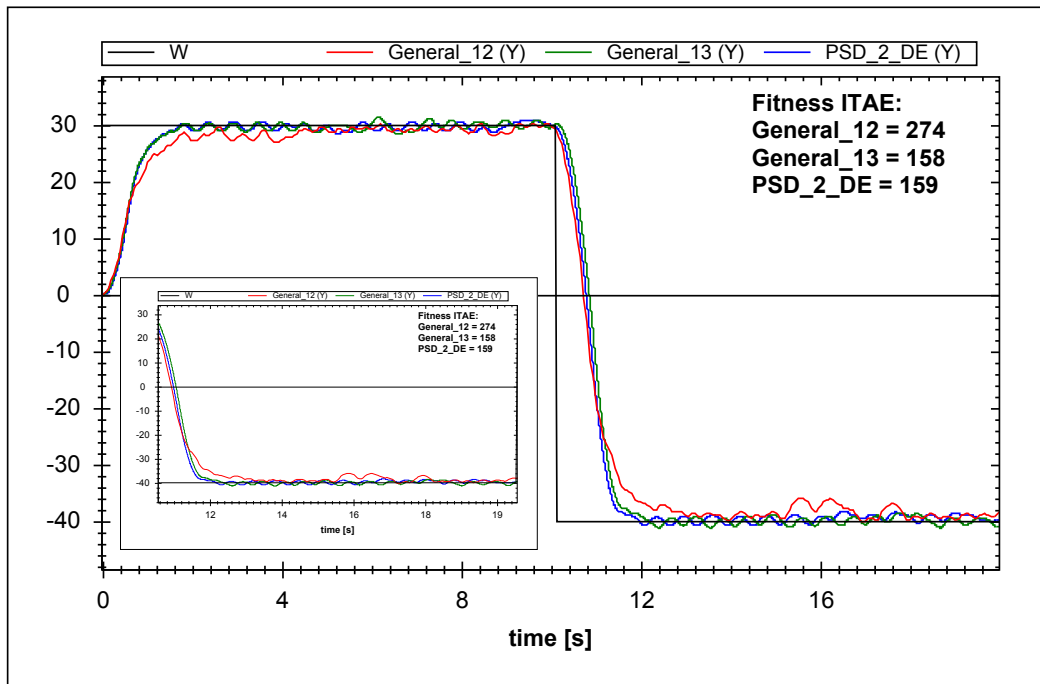
$$U_k = U_{k-1} + (((- (E_k + (Y_{k-1} - 14, 4044473368695))) + (- (((E_{k-2} * (-1, 69526341729436)) - (E_k * 41, 3497823420514)) - (E_{k-1} * (-38, 0841704174065)))) * (E_k - (-71, 1561559627034))))))$$

Rovnice byla ponechána v původním tvaru v jakém jej vrátila *TLTE*. V průběhu optimalizace již byla do rovnice uměle přidána setrvačnost prvního řádu ve formě členu U_{k-1} .

Výsledkem optimalizace ve 22. generaci *TLTE* je řídicí rovnice *obecného regulátoru General_13* ve tvaru:

$$U_k = U_{k-1} + (((- (((- ((E_{k-1} * 46, 1238341489317) - ((E_k * (-6, 86712128949357)) * (-6, 84720209198847)))) * (-50, 3217781662577)) - 0, 723919624277033)))$$

Rovnice byla ponechána v původním tvaru v jakém jej vrátila *TLTE*. V průběhu optimalizace již byla do rovnice uměle přidána setrvačnost prvního řádu ve formě členu U_{k-1} , stejně jako v případě optimalizace *General_12*.



General_12: Obecný regulátor (struktura a parametry optimalizovány *TLTE* pro $T = 0,1s$, kritéria hodnocení: ITAE + maximální překmit „y“, 7.generace *TLTE*, setrvačnost 1.řádu, pásmo necitlivosti 1% ($\nabla y(kT) = 0,1$), filtr 2.řádu, omezená pravidla: $Y_k, \dots, Y_{k-2}; E_k, \dots, E_{k-2}$), **General_13:** Obecný regulátor (struktura a parametry optimalizovány *TLTE* pro $T = 0,01s$, kritéria hodnocení: ITAE + maximální překmit „y“, 22.generace *TLTE*, setrvačnost 1.řádu, pásmo necitlivosti 1% ($\nabla y(kT) = 0,1$), filtr 2.řádu), **PSD_2_DE:** PSD regulátor (parametry optimalizovány *DE* pro $T = 0,01s$, kritéria hodnocení: ITAE + maximální překmit „y“, 1500.generací *DE*, pásmo necitlivosti 1% ($\nabla y(kT) = 0,1$), filtr 2.řádu)

Obrázek 21: Porovnání obecných regulátorů se setrvačností prvního řádu a PSD regulátorů, s pásmem necitlivosti – kmitavá proporcionální soustava bez dopravního zpoždění

Kritériální funkce ve všech případech hodnotila dle kritéria *ITAE*, ke kterému byla přidána penalizace překmitu regulované veličiny. Do řídicích rovnic obecných regulátorů byl uměle přidán setrvačný člen 1.řádu, pásmo necitlivosti bylo ve všech případech nastaveno na 1% při zpětné diferenci $\nabla y(kT) = 0,1$, všechny regulátory používaly filtr regulační odchylky 2.řádu.

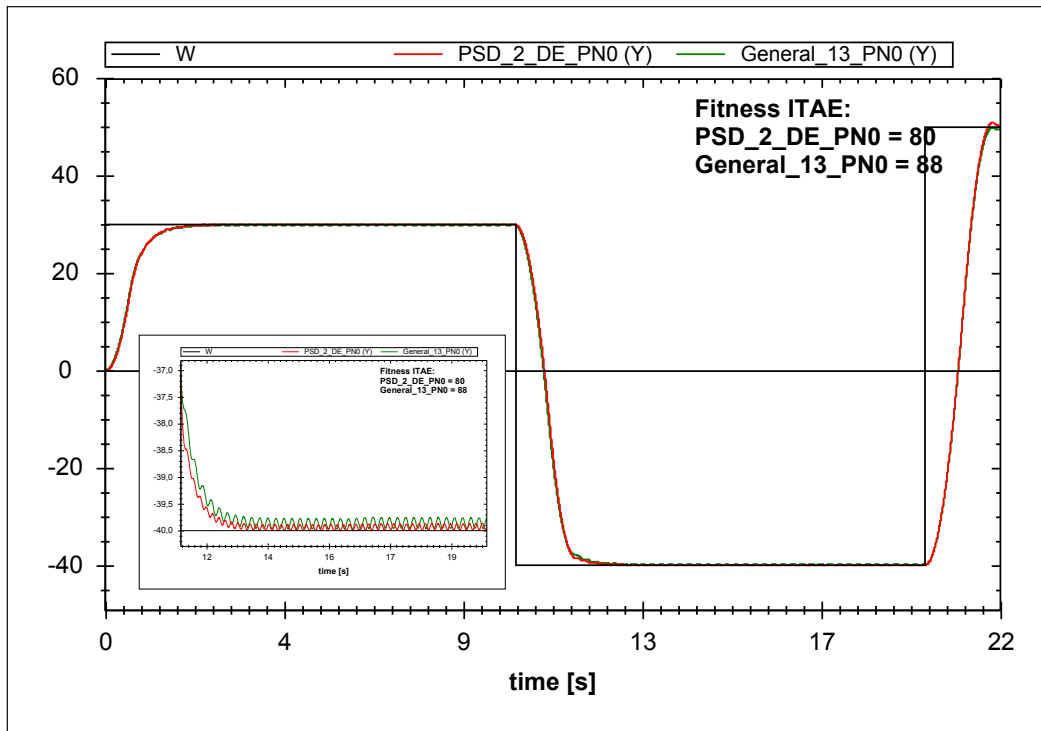
Výsledné regulační pochody je možno porovnat na obrázku 21. Na tomto obrázku jsou ukázány tři regulační pochody. V pravém horním rohu obrázku je zobrazeno hodnocení *ITAE*, v levém dolním rohu je ukázán detail regulačních pochodů v intervalu od $\approx 11s$ do $\approx 19s$.

Na obrázku 22 jsou ukázány regulační pochody regulátorů *General_13* a *PSD_2_DE* s vypnutými pásmy necitlivosti. Tyto regulátory jsou označeny *General_13_PN0* a *PSD_2_DE_PN0*. V pravém horním rohu obrázku je zobrazeno hodnocení *ITAE*, v levém dolním rohu je ukázán detail regulačních pochodů v intervalu od $\approx 11s$ do $\approx 19s$.

Na obrázku 23 je výsledek optimalizace řídicí rovnice obecného regulátoru *General_12* v 10.generaci *TLTE*. Průběh regulačního pochodu je ukázán v intervalu $(0, 90)$ sekund, přičemž interval změny žádané hodnoty se na počátku mění periodicky po 5s a postupně se snižuje až na 1s. Změny hodnoty žádané veličiny jsou náhodné.

Na tomto obrázku je vidět kmitání akčního zásahu, jenž je shora a zdola omezeno daným rozsahem regulátoru, který je nastaven v intervalu ± 200 . Toto kmitání je způsobeno tím, že pro hodnocení kvality regulace během optimalizace nebylo použito kritéria, které penalizuje kmitání akčního zásahu regulátoru.

Rovnice tohoto regulátoru má tvar:



General_13_PN0: Obecný regulátor (struktura a parametry optimalizovány *TLTE* pro $T = 0,01s$, kritéria hodnocení: ITAE + maximální překmit „y“, 22.generace *TLTE*, setrvačnost 1.řádu, filtr 2.řádu),

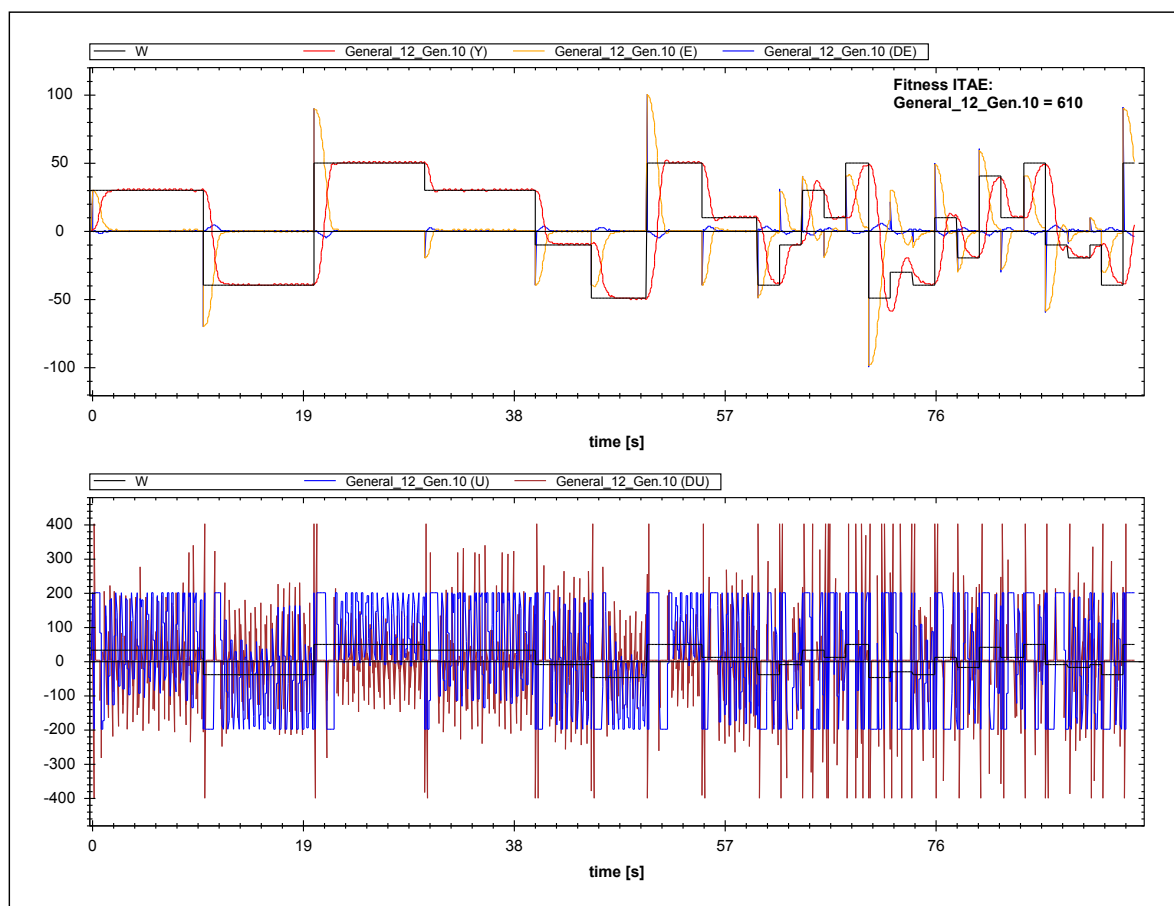
PSD_2_DE_PN0: PSD regulátor (parametry optimalizovány *DE* pro $T = 0,01s$, kritéria hodnocení: ITAE + maximální překmit „y“, 1500.generací *DE*, filtr 2.řádu)

Obrázek 22: Porovnání obecných regulátorů se setrvačností prvního řádu a PSD regulátorů, bez pásma necitlivosti – kmitavá proporcionální soustava bez dopravního zpoždění

$$\begin{aligned}
 U_k = & U_{k-1} + (((((E_k \cdot (-0,663520535763126)) - (((E_k \cdot (-19,4200644798541)) - \\
 & (E_k \cdot 62,8320479267121)) - (E_{k-1} \cdot (-74,5999125652272))) \cdot \\
 & 9,69180579919801)) - (E_{k-1} \cdot 25,2196416841911)) + \\
 & (E_k \cdot 7,51412533075408)) - (-6,26645661250988))
 \end{aligned}$$

Zobrazená rovnice byla ponechána v původním tvaru v jakém jej vrátila *TLTE* po 10.generaci.

Z časového hlediska byla optimalizace přerušena po 11.generacích *TLTE* (10,5 hodin CPU času 3,2GHz CPU). Za tuto dobu bylo vyčísleno 351,7 tis. generací *DE* s průměrných počtem 1056,3 generací *DE* na jednoho jedince *TE*.



Horní graf: průběh regulačního pochodu, **Dolní graf:** průběh akčních zásahů, **červená (Y):** regulovaná veličina, **oranžová (E):** regulační odchylka, **modrá (DE):** derivace regulační odchylky, **modrá (U):** akční zásah regulátoru, **hnědá (DU):** derivace akčního zásahu, **černá (W):** žádaná hodnota regulované veličiny

Obrázek 23: Průběh regulačního pochodu regulátoru *General_12* v 10.generaci *TLTE*

7 ZÁVĚR

Mým záměrem při tvorbě této práce bylo proniknout do podstaty řídicích algoritmů regulátorů a pokusit se optimalizovat řídicí algoritmy obecných regulátorů pro danou soustavu a žádaný průběh regulačního pochodu. Dílčím úkolem bylo vytvoření části aplikace, jenž bude pro zadanou soustavu a řídicí rovnici regulátoru simulovat regulační pochod, který poté dle zadaných kritérií ohodnotí. Dalším úkolem bylo vytvoření evolučního algoritmu, který je schopen optimalizovat parametry PSD regulátoru. Pro splnění tohoto úkolu byla implementována dobře známá metoda *diferenciální evoluce*.

Hlavní náplní této práce a současně jedním z největších přínosů je optimalizace struktur řídicích rovnic obecných regulátorů pro danou soustavu a požadavky na průběh regulačního pochodu. Pro splnění tohoto cíle byl vytvořen originální evoluční algoritmus nazvaný *transplantační evoluce* a její rozšíření nazvané *dvoufázová transplantační evoluce*. Tyto typy evolučních algoritmů jsou popsány v kapitolách 2 a 3.

Klíčovou vlastností a zároveň největší výhodou transplantační evoluce je schopnost vytvořit model řešené úlohy, který je srozumitelný uživateli a je použitelný i mimo evoluční metodu. V tomto ohledu tedy svými schopnostmi transplantační evoluce předčí neuronové sítě, protože model vytvořený neuronovou sítí není použitelný mimo ni. Další výhodou tentokrát v porovnání s gramatickou evolucí je možnost vložit řešení do transplantační evoluce a toto řešení dále optimalizovat bez nutnosti složitě vytvářet genotyp jedince. Vzhledem k tomu, že jedinec transplantační evoluce je reprezentován jen fenotypem, bylo možno vytvořit pokročilé metody křížení a mutace, které jsou plně řízeny pravidly gramatiky, narozdíl od gramatické evoluce, v níž křížení a mutace probíhá na úrovni genotypu a gramatika se používá až v další fázi, během translace genotypu na fenotyp. Další výhodou transplantační evoluce v porovnání s gramatickou evolucí je možnost změny pravidel v průběhu evoluce, bez totální destrukce již vygenerovaných řešení. Vzhledem k tomu, že požadavkem optimalizace na výsledný model je získání strukturálně minimálního (optimálního) řešení, je jedním z největších přínosů transplantační evoluce možnost použití *přímé aritmetické redukce modelu*, která zajistí redukci jeho redundantních částí, což má mj. přímý vliv na zkrácení doby optimalizace. Přínosem dvoufázové transplantační evoluce, která byla inspirována nedávno vytvořeným algoritmem dvoufázové gramatické evoluce je přímá a efektivní optimalizace abstraktních numerických parametrů obsažených ve struktuře řídicí rovnice regulátoru, jejíž správné nastavení je rozhodující pro zajištění kvalitního průběhu regulačního pochodu a pro nalezení strukturálně optimálního řešení. Vzhledem k citlivosti řídicí rovnice regulátoru na nastavení parametrů, je tato činnost v gramatické, nebo transplantační evoluci naprosto neřešitelná.

Pro zajištění optimálního chodu druhé fáze optimalizace byl vytvořen meta-evoluční algoritmus, jehož cílem byla optimalizace parametrů diferenciální evoluce, která je použita pro optimalizaci parametrů řídicích rovnic regulátorů jako součást dvoufázové transplantační evoluce, nebo samostatně pro optimalizaci parametrů PSD regulátoru. Popisu *meta-evoluce* se věnuje kapitola 4.

V průběhu experimentů bylo zjištěno, že standardní schéma diferenciální evoluce DE/Rand/1 (DE/Best/1) v určitých situacích nekonverguje k očekávaným hodnotám, proto bylo vytvořeno opět původní schéma, jenž bylo nazváno *DE/Rand/1mod*, případně *DE/Rand/1plus* (*DE/Best/1mod*, *DE/Best/1plus*). Přínos tohoto schématu je sice diskutabilní, jelikož po vytvoření meta-evolučního algoritmu a optimalizování parametrů diferenciální evoluce vykazuje podobné výsledky jako původní schéma, nicméně i přesto bylo toto schéma v průběhu optimalizací používáno. Uvedené schéma je popsáno v kapitole 5.3.

Pro ohodnocení kvality regulačních pochodů byly vytvořeny opět originální metody, které byly kombinovány s běžnými integrálními kritérii velikosti regulační plochy. Tyto kritéria jsou schopny plně nahradit kritérium *ITAE*, případně jsou použitelné jako doplňková kritéria ohodnocení kvality regulačního pochodu ke kritériím *ITAE*, apod. pro zajištění žádaného průběhu regulované veličiny.

Výsledky optimalizací byly simulačně porovnány s *PSD regulátory*, nastavenými pomocí kla-

sických výpočetních i evolučních metod. Ve všech případech optimalizace se podařilo vygenerovat kvalitnější řídicí rovnici, než rovnice *PSD regulátorů*, případně její známá modifikace v *Takahashiho* tvaru. Nově vygenerované řídicí algoritmy ve všech případech vykazovaly kvalitnější průběh regulačního pochodu současně se sníženou dobou regulace.

Výsledkem experimentů je jednoznačný závěr, že optimalizací řídicích rovnic obecných regulátorů je možno získat kvalitnější průběh regulačního pochodu, než jaký nabízejí klasické *PSD regulátory* a jejich známé modifikace. Časová náročnost pro nalezení rovnice *obecného regulátoru* je ovšem mnohem větší, než nalezení parametrů *PSD regulátoru*.

V závěru práce je uvedeno doporučení pro další vývoj v tomto oboru, jenž se týká paralelních a distribuovaných výpočtů a týká se hlavně zkrácení doby optimalizace.

Hlavní přínos práce je shrnut do následujících bodů:

- Byly vytvořeny původní metody *transplantační evoluce* a její modifikace *dvoufázová transplantační evoluce*, které jsou obecně využitelné pro různé optimalizační problémy, (viz. kapitoly 2 a 3),
- Byla vytvořena původní metoda přímé aritmetické redukce modelu řešení (*ART*), (viz. kapitola 3.1),
- Byla vytvořena nová metoda křížení spojováním částí jedince, (viz. kapitola 2.3),
- Byly vytvořeny gramatikou řízené metody křížení a mutace, (viz. kapitoly 2.3, 2.4),
- Byly vytvořeny původní kritéria, hodnotící kvalitu regulace, (viz. kapitola 5.2),
- Byly vytvořeny unikátní kritéria hodnocení kvality regulačního pochodu v případě chyby v průběhu regulace, (viz. kapitola 5.2.1),
- Bylo vytvořeno nové schéma diferenciální evoluce, (viz. kapitola 5.3),
- Byly porovnány regulační pochody *PSD regulátorů*, *Takahashiho regulátorů* a obecných regulátorů pro různé možnosti nastavení, různé regulované soustavy a omezující podmínky. Tyto experimenty jsou uvedeny v kapitole 6.
- Pro všechny vybrané soustavy se podařilo nalézt takovou řídicí rovnici obecného regulátoru, která svou strukturou a parametry dokáže zajistit kvalitnější průběh regulačního pochodu, než rovnice *PSD regulátoru*, případně její modifikace v *Takahashiho* tvaru.

Pro zajištění vysoké typografické kvality byla práce vysázena v programu $\text{\LaTeX}$ a všechny obrázky byly vlastnoručně nakresleny ve vysokém rozlišení. Všechny experimenty, optimalizační úlohy a výstupy simulací byly prováděny v programu, který jsem vytvořil pro tyto účely v jazyku $\text{C\#}$.NET

8 LITERATURA

- [1] BÄCK, T.; HAMMEL, U.; SCHWEFEL, H. P. Evolutionary computation : comments on the history and current state. In . *IEEE Transactions on Evolutionary Computation*. 1. [s.l.] : [s.n.], 1997. s. 3–17. ISSN 1089–778X.
- [2] BACKUS, J. W., et al. *ALGOL-like Languages*. Volume 1. Cambridge, MA, USA : Birkhauser Boston Inc., 1997. Revised report on the algorithmic language ALGOL 60, s. 19–49. ISBN 0–8176–3880–6.
- [3] BALÁTĚ, J. *Automatické Řízení*. 2. přepracované vydání. Praha : BEN, 2004. 664 s. ISBN 80-7300-148-9.
- [4] BLICKLE, T.; THIELE, L. *A comparison of selection schemes used in evolutionary algorithms*. *Evolutionary Computation*. 1996, Volume 4, Issue 4, s. 361–394. ISSN 1063–6560.
- [5] CHAKRABORTY, Uday K. *Advances in Differential Evolution*. Berlin: Springer, 2008. 319 s. ISBN 978–3–540–68827–3.
- [6] CHAKRABORTY, U. K.; DEB, K.; CHAKRABORTY, M. Analysis of Selection Algorithms : A Markov Chain Approach. *Evolutionary Computation*. 1996, Volume 4, Issue 2, s. 133–167. ISSN 1063–6560.
- [7] DAS, Swagatam; ABRAHAM, Ajith; KONAR, Amit. *Metaheuristic Clustering*. Berlin Heidelberg : Springer-Verlag, 2009. 254 s. ISBN 978–3–540–92172–1.
- [8] DEMPSEY, I.; O’NEILL, M.; BRABAZON, A. Meta-grammar constant creation with grammatical evolution by grammatical evolution. In . *Genetic and Evolutionary Computation Conference*. Washington DC, USA : GECCO, 2005. s. 1665–1671. ISBN 1-59593-010-8.
- [9] GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1. [s.l.] : Addison–Wesley Professional, 1989. 432 s. ISBN 0201157675, ISBN-13: 978–0201157673.
- [10] HAVLÍN, V. *Preferenční křížení v evolučních algoritmech*. Praha, 2009. 70 s. Bakalářská práce. České vysoké učení technické v Praze, Fakulta elektrotechnická.
- [11] HYNEK, J. *Genetické algoritmy a genetické programování*. [ČR] : Grada Publishing a.s., 2008. 182 s. ISBN 8024726955
- [12] JAKOUBEK, P. Identifikace parametrů stabilních nekmitavých soustav. *Automatizace*. 2009, 52, 2, s. 84–89. Dostupný také z WWW: <http://www.automatizace.cz/download/article/2444_a0209_84.pdf>.
- [13] KOZA, J. R. *Genetic Programming : On the Programming of Computers by Means of Natural Selection*. USA : The MIT Press, 1992. 835 s. ISBN 0–262–11170–5, ISBN-13: 978–0–262–11170–6.
- [14] KRASNOGOR, N., et al. *Studies in Computational Intelligence : Nature Inspired Cooperative Strategies for Optimisation*. Berlin Heidelberg : Springer-Verlag, 2009. Genetic Algorithm and Advanced Tournament Selection Concept, s. 187–196. ISBN 978–3–642–03210–3.

- [15] KŘENA, B. *Metody analýzy objektově orientovaných Petriho sítí*. Brno : Vysoké učení technické v Brně, 2004. 128 s. Dizertační práce. Fakulta informačních technologií, Vysoké učení technické v Brně.
- [16] LAMPINEN, J.; ZELINKA, I. Mixed integer-discrete-continuous optimization by differential evolution: Part 1: the optimization method. In . *5th International Mendel Conference on Soft Computing*. Brno: Mendel, 1999. s. 71–76. ISBN 80–214–11.
- [17] LEPS, M. *Moderní metody optimalizace* [online]. 25. Listopadu 2009 [cit. 2010-06-10]. Dostupné z WWW: <<http://klobouk.fsv.cvut.cz/~leps/teaching/mmo/>>.
- [18] LIPTÁK, Béla G. *Process Control and Optimization : Instrument Engineers' Handbook*. 4th edition. Boca Raton : CRC Press, 2006. 2386 s. ISBN 0–8493–1081–4.
- [19] MITCHELL, M. *An Introduction to Genetic Algorithms*. Cambridge, Massachusetts : The MIT Press, 1998. 221 s. ISBN 0–262–63185–7, ISBN–13: 978–0–262–63185–3.
- [20] NOHEJL, A. *Grammatical Evolution*. [s.l.], 2009. 127 s. Bakalářská práce. Charles University in Prague, Faculty of Mathematics and Physics.
- [21] O'NEILL, M; BRABAZON, A. Grammatical Differential Evolution. In . *International Conference on Artificial Intelligence, ICAI*. Volume 1. Las Vegas, Nevada, USA : CSREA Press, 2006. s. 231–236. ISBN 1-932415-96-3.
- [22] O'NEILL, M.; RYAN, C. *Grammatical Evolution : Evolutionary Automatic Programming in an Arbitrary Language*. 1st edition. [s.l.] : Springer, 2003. 160 s. ISBN 1-4020-7444-1, ISBN-13: 978-1402074448.
- [23] O'NEILL, M. *Grammatical-evolution.org* [online]. Last updated 01/04/2010 [cit. 2010-05-02]. Dostupné z WWW: <<http://www.grammatical-evolution.org>>.
- [24] OŠMERA, P. *Advances in Computational Algorithms and Data Analysis : Lecture Notes in Electrical Engineering*. Netherlands : Springer, 2009. Two Level Parallel Grammatical Evolution, s. 509–525. ISBN 978-1-4020-8918-3.
- [25] PAVLOVIČ, J. *Multikriteriální hybridní evoluční algoritmy pro výběr a optimalizaci dekontaminačních technologií* [online]. Brno : Masarykova univerzita, 2006. 30 s. Teze disertační práce. Masarykova univerzita. Dostupné z WWW: <http://is.muni.cz/th/4035/fi_r/teze_extended.pdf>
- [26] POPELKA, O. *Použití evolučních a genetických algoritmů v ekonomických aplikacích*. Brno, 2009. 133 s. Disertační práce. Mendelova zemědělská a lesnická univerzita v Brně, Provozně ekonomická fakulta.
- [27] PRICE, Kenneth V.; STORN, Rainer M.; LAMPINEN, Jouni A. *Differential Evolution: A Practical Approach to Global Optimization*. Berlín: Springer-Verlag, 2005. 535 s. ISBN 3–540–20950–6.
- [28] QIN, A. K.; HUANG, V. L.; SUGANTHAN, P. N. Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. In *IEEE: transactions on evolutionary computation*. vol. 13. New York: Institute of Electrical and Electronics Engineers, 2009. s. 398–417. ISSN 1089–778X.
- [29] RYAN, C.; COLLINS, J. J.; O, M. Grammatical Evolution : Evolving Programs for an Arbitrary Language. In . *EuroGP : Proceedings of the First European Workshop on Genetic Programming*. London, UK : Springer-Verlag, 1998. s. 83–96. ISBN 3–540–64360–5.

- [30] Ryan C., O'Neill M., Collins J.J. *Grammatical Evolution: Solving Trigonometric Identities*. In proceedings of Mendel 1998: 4th International Mendel Conference on Genetic Algorithms, Optimisation Problems, Fuzzy Logic, Neural Networks, Rough Sets. June 24–26 1998, Brno, Czech Republic, pages 111–119. ISBN 80–214–1199–6
- [31] ŠVARC, I.; ŠEDA, M.; VÍTEČKOVÁ, M. *Automatické řízení*. první vydání. Brno : Akademické nakladatelství CERM, s.r.o., 2007. 324 s. ISBN 978–80–214–3491–2.
- [32] WEISSER, R. *Evoluční optimalizace řídicích algoritmů*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2010. 157 s. Vedoucí dizertační práce prof. Ing. Pavel Ošmera, CSc.
- [33] WEISSER, R., *Optimization of Structure and Parameters of Controller*. In International Conference on Soft Computing Applied in Computer and Economic Environments ICSC. 8th. Hodonín : Evropský polytechnický institut, s.r.o., 2010. s. 159–166. ISBN 978-80-7314-201-8.
- [34] WEISSER, R., et al, *Transplant Evolution for Optimization of General Controllers*. In European Conference on Modelling and Simulation. 24th. Kuala Lumpur (Malaysia) : ECMS 2010. s. 250 – 260. ISBN 978–0–9564944–0–5.
- [35] WEISSER, R. Parallel Grammatical Evolution : Solving Trigonometric Identities with Backward Processing. In *International Conference on Soft Computing MENDEL*. Brno : MENDEL, 2005. CD version. ISBN 80–214–2961–5.
- [36] WEISSER, R., OŠMERA, P., MATOUŠEK, R., Transplant Evolution with Modified Schema of Differential Evolution : Optimization Structure of Controllers. In *International Conference on Soft Computing MENDEL*. Brno : MENDEL, 2010.
- [37] YANG, Z.; TANG, K.; YAO, X. Self-adaptive Differential Evolution with Neighborhood Search. In *IEEE Congress on Evolutionary Computation*. Hong Kong, China: IEEE, 2008. s. 1110–1116. ISBN 978–1–4244–1822–0.
- [38] ISO/IEC 14977:1996(E). *Information technology : Syntactic metalanguage–Extended BNF*. [s.l.] : ISO, 1996. 24 s. Dostupné z WWW: <<http://standards.iso.org/ittf/licence.html>>
- [39] *Rex Controls* [online]. 2000 [cit. 2010–05–18]. Dostupné z WWW: <<http://www.rexcontrols.cz>>.
- [40] ZELINKA, I., et al. *Evoluční výpočetní techniky : principy a aplikace*. 1. vyd. Praha : BEN, 2009. 536 s. ISBN 978–80–7300–218–3.

9 SEZNAM VLASTNÍCH PUBLIKACÍ

WEISSER, R., OŠMERA, P., MATOUŠEK, R., Transplant Evolution with Modified Schema of Differential Evolution : Optimization Structure of Controllers. *In International Conference on Soft Computing MENDEL*. Brno : MENDEL, 2010.

WEISSER, R., Optimization of Structure and Parameters of Controller. *In International Conference on Soft Computing Applied in Computer and Economic Environments ICSC. 8th.* Hodonín : Evropský polytechnický institut, s.r.o., 2010. s. 159–166. ISBN 978-80-7314-201-8.

WEISSER, R., OŠMERA, P., ŠEDA, M., KRATOCHVÍL, O. Transplant Evolution for Optimization of General Controllers. *In European Conference on Modelling and Simulation. 24th.* Kuala Lumpur (Malaysia) : ECMS 2010. s. 250 – 260. ISBN 978-0-9564944-0-5.

ŠEDA, M., MATOUŠEK, R., OŠMERA, P., ŠANDERA, WEISSER, R.,: Combined Heuristic Approach to Resource-Constrained Project Sceduling Problem. 12 pp. (*v tisku jako kapitola v knize vydané nakladatelstvím Springer-Verlag*)

ŠEDA, M., MATOUŠEK, R., OŠMERA, P., ŠANDERA, WEISSER, R.,: A Knapsack-Problem-Based Shifting Heuristic Algorithm for Project Scheduling with Limited Resources. 7 pp. (*v tisku jako speciální číslo časopisu IAENG Journal*)

WEISSER, R. Parallel Grammatical Evolution : Solving Trigonometric Identities with Backward Processing. *In International Confeence on Soft Computing MENDEL*. Brno : MENDEL, 2005. CD version. ISBN 80–214–2961–5.

ČECHMAN, J., WEISSER, R. Shape Optimization of 3D Embryo Model by Evolutionary Computing. *In East West Fuzzy Colloquium 2004*. Zittau: Hochschule Zittau, 2004. s. 213-218. ISBN: 3-98-080089-5-5.

Ing. Roman Weisser

Curriculum Vitae

Osobní údaje:

Datum narození:	8.12. 1979
Adresa trvalého bydliště:	Polní 966, 735 14 Orlová
Kontaktní telefon:	(+420) 608 034 494
E-mail:	roman.weisser@gmail.com
Národnost:	česká
Rodinný stav:	svobodný

Pracovní zkušenosti:

Květen 2009 – Červenec 2009: **Cyber Stream Technology s.r.o.**
programátor .NET

Duben 2009: **Xacti CZ, s.r.o.**
.NET vývojář / analytik

Únor 2007 – Březen 2009: **Merz, spol. s.r.o.**
Programátor / projektant

Vzdělání:

2003 – dosud: **Vysoké učení technické v Brně – fakulta strojního inženýrství**
postgraduální studium – ústav mechaniky těles, mechatroniky a biomechaniky

disertační práce: Evoluční optimalizace řídicích algoritmů

1998 – 2003: **Vysoké učení technické v Brně – fakulta strojního inženýrství**
magisterské studium – ústav automatizace a informatiky

kvalifikace: Inženýr v oboru informatika a automatizace

diplomová práce: Software pro identifikaci objektů realizovaný v programovacím jazyku C++ (diplomová práce byla součástí vědeckého výzkumu „Netradiční metody studia komplexních a neurčitých systémů“)

1994 – 1998: **Střední škola strojní, tř. Osvobození 60 /1111, Karviná**

Certifikace:

- Interní auditor ISO 9001:2008

10 ABSTRACT

The dissertation thesis deals with Evolution optimization of control algorithms. The first part of the thesis describes the principles and partial methods of evolution optimization methods especially those used in two-level transplant evolution method. Later the grammatical evolution method is described, which modified algorithm became impulse for creation of transplant evolution method. The transplant evolution method and its two-level modification are new evolutionary algorithms proposed in this work, which were used for optimization of structure and parameters of general controllers control algorithms. The transplant evolution algorithm and its extended two-level modification are described in detail in next chapters.

The proper settings of evolutionary algorithms are important for minimization the time of optimization and for finds results approaching the global optimum. For proper setting the parameters of differential evolution was created meta-evolution algorithm that is described in chapter named meta-evolution.

The basic concepts of control, chosen methods of system identification and controller parameters settings are described in next part. This part describes algorithms of digital controllers and some specific methods uses in digital control.

The demonstrations of control algorithm optimizations of various types of controllers are showed in experimental part. The optimized algorithms of general controllers are compared with various types of PSD controllers which were set by various algebraic methods or differential evolution for various models of systems.

In the conclusion of this work is stated a recommendation for further development of evolutionary optimization of controllers are focusing on parallel and distributed computing.