

Symbolické řízení

Bc. David Bijok

Diplomová práce
2006



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav řízení procesů

akademický rok: 2005/2006

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. David BIJOK**

Studijní program: **N 2807 Chemické a procesní inženýrství**

Studijní obor: **Automatizace a řídicí technika**

Téma práce: **Symbolické řízení**

Zásady pro vypracování:

Cílem práce je použít metody symbolické regrese na identifikaci systémů a návrh struktury regulátorů. Obsahem práce bude vytvoření množiny simulací s cílem identifikovat vybrané reprezentativní dynamické systémy a případný návrh regulátorů. Jako metoda symbolické regrese bude použito analytické programování.:

- 1. vypracovat přehled problematiky identifikace a syntézy dynamických systémů**
- 2. vybrat vhodné již řešené příklady**
- 3. vypracovat alternativní řešení pomocí analytického programování a vybraného algoritmu z SOMA, DE, SA a GA**
- 4. Provést závěr**

Rozsah práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

[1] ZELINKA, Ivan. Umělá inteligence I. Volume 1. Zlín: Vutum, Brno, 1998. 126p. ISBN 80-214-1163-5.

[2] ZELINKA, Ivan. Umělá inteligence /kap.6 "Diferenciální evoluce", Academia, 33p.

[3] Kvasnička V., Pospíchal J., Tiňo P. 2000, Evolučné algoritmy, STU Bratislava, ISBN 85-246-2000,2000

[4] ZELINKA, Ivan, New Optimization Techniques in Engineering /kap.7 "SOMA - Self Organizing Migrating Algorithm, Springer-Verlag

[5] Koza J.R. 1998, Genetic Programming, MIT Press, ISBN 0-262-11189-6,1998

[6] Koza J.R., Bennet F.H., Andre D., Keane M. 1999, Genetic Programming III, Morgan Kaufmann pub., ISBN 1-55860-543-6, 1999

Vedoucí diplomové práce:

doc. Ing. Ivan Zelinka, Ph.D.

Ústav aplikované informatiky

Datum zadání diplomové práce:

14. února 2006

Termín odevzdání diplomové práce:

26. května 2006

Vě Zlíně dne 25. února 2006

prof. Ing. Vladimír Vašek, CSc.
pověřený děkan



prof. Ing. Petr Dostál, CSc.
ředitel ústavu

ABSTRAKT

Identifikovat regulovaný systém potřebujeme k návrhu a nastavení regulátoru. Vytvořit matematický model lze více způsoby. Tato práce si dává za cíl ověřit možnost vytvoření matematického modelu systému z naměřených dat (přechodová charakteristika) pomocí symbolické regrese (analytického programování) a evolučního algoritmu (SOMA). Simulace byly provedeny pro 5 testovacích systémů. Výsledky ukazují, že identifikace tímto způsobem dosahuje sice pro každý testovaný systém lehce odlišných, ale i přesto dobrých výsledků.

Klíčová slova: Identifikace, Symbolická regrese, Analytické programování, Evoluční algoritmy, SOMA

ABSTRACT

We need to identify the regulate system to design and to set the regulator. We can create mathematical model by several ways. The goal of this work is verifying the possibility of creation of mathematical model from recorded dates (plant response to step input) by help of symbolic regression (Analytic programming) and evolution algorithms (SOMA). Simulations were performed for 5 test systems. Results say, that identification by this way, shows good result, but for each test system with small difference.

Keywords: Identification, Symbolic regression, Analytic programming, Evolution algorithms, SOMA

Poděkování

Děkuji vedoucímu diplomové práce panu Doc. Ivanovi Zelinkovi, Ph.D. za odborné vedení, rady a připomínky, které mi poskytoval při řešení této práce.

OBSAH

ÚVOD	8
I TEORETICKÁ ČÁST	9
1 IDENTIFIKACE	10
1.1 KLASIFIKACE MATEMATICKÝCH MODELŮ	10
1.2 EXPERIMENTÁLNÍ METODY IDENTIFIKACE	11
1.3 VYHODNOCOVÁNÍ PŘECHODOVÝCH CHARAKTERISTIK	14
1.3.1 Aproximace soustavou prvního řádu bez dopravního zpoždění	14
1.3.2 Aproximace statických soustav vyšších řádů	16
2 ŘEŠENÉ PŘÍKLADY IDENTIFIKACE	17
3 EVOLUČNÍ ALGORITMY	21
3.1 GA – GENETICKÝ ALGORITMUS	21
3.1.1 Kódování a fitness funkce	21
3.1.2 Reprodukce.....	23
3.2 SA – SIMULOVANÉ ŽIHÁNÍ.....	23
3.3 DE – DIFERENCIÁLNÍ EVOLUCE.....	25
3.4 SOMA – SAMO-ORGANIZUJÍCÍ SE MIGRAČNÍ ALGORITMUS	27
3.5 POČET VYHODNOCENÍ ÚČELOVÉ FUNKCE PRO JEDNOTLIVÉ ALGORITMY	30
4 AP - ANALYTICKÉ PROGRAMOVÁNÍ	32
4.1 MNOŽINA DISKRÉTNÍCH HODNOT.....	32
4.2 OBECNÝ FUNKČNÍ PROSTOR.....	32
4.3 ZÁKLADNÍ PRINCIPY	33
4.4 NALEZENÍ VÝSLEDNÉ FUNKCE	35
4.5 IMPLEMENTACE AP	35
4.5.1 MainAP, množina funkcí	36
4.5.2 PickUp funkce	36
4.5.3 GiveFunction	36
4.5.4 CostValue	37
II PRAKTICKÁ ČÁST	38
5 VYPRACOVÁNÍ V AP	39
6 VÝSLEDKY SIMULACÍ	41
ZÁVĚR	44
SEZNAM POUŽITÉ LITERATURY	45
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	46
SEZNAM OBRÁZKŮ	47
SEZNAM TABULEK	49

SEZNAM PŘÍLOH.....	50
---------------------------	-----------

ÚVOD

V praxi se setkáváme často s požadavkem na znalost matematického modelu systému. Jeho znalost potřebujeme pro navržení struktury regulátoru a nastavení jeho parametrů.

Ve většině případů je analytické zjišťování modelu ,identifikace, obtížné, nebo nemožné. Často známe jen výstup ze systému na základě známého vstupu.

Tato práce si dává za cíl ověřit možnost vytvoření matematického modelu pomocí symbolické regrese (analytického programování) a evolučních algoritmů (SOMA). Výstupem ze simulací jsou přechodové charakteristiky, které popisují odezvu systému na jednotkový skok na vstupu.

V první části diplomové práce je stručný přehled experimentální deterministické identifikace. V další části je příklad identifikace systému jak ho řešil Koza [2]. V diplomové práci je přehled principů základních evolučních algoritmů, jako jsou GA, SA, DE nebo SOMA. Práce dále podává stručný popis a principy Analytického programování.

V praktické části jsou prezentovány výsledky identifikace testovaných systémů.

V práci je použit software Mathematica firmy Wolfram (www.wolfram.com) jako programovací prostředí pro simulace.

I. TEORETICKÁ ČÁST

1 IDENTIFIKACE

Identifikací v širším slova smyslu se rozumí proces ztotožňování poznatků a vědomostí o zkoumaném objektu se skutečností. Jedná se tedy o poznávací proces, který si je možno představit jako orientovanou interakci mezi poznávaným objektem (skutečností) a poznávacím subjektem (pozorovatelem). Výsledkem poznávacího procesu je určité poznání o poznávaném objektu, které poznávající subjekt formuluje do jistých vět, pouček, matematických vztahů apod. [1]

Dá se také říct, že identifikace je redukce systému na jeho matematický model.

1.1 Klasifikace matematických modelů

Matematické modely dělíme podle jejich charakteru na:

- Statické – vazbu mezi vstupními a výstupními veličinami reprezentují algebraické rovnice. Popisuje ustálené stavy.
- Dynamické – vazbu mezi vstupy a výstupy vyjadřují diferenciální nebo diferenční rovnice. Modely statické dostaneme obecně z modelu dynamického pro limitu $t \rightarrow \infty$.

Podle toho, jestli koeficienty rovnic (obecně parametry dynamického modelu) jsou závislé na čase:

- Časově závislé (t-variantní)
- Časově nezávislé (t-invariantní)

Z hlediska linearity:

- Lineární – operace jsou jednodušší než s nelineárním modelem.
- Nelineární – podle možností přistupujeme k linearizaci.

Podle popisu vazeb mezi veličinami:

- Spojitý – pouze spojité změny veličin
- Diskrétní – změny veličin jen v diskrétních časových okamžicích.

Podle charakteru vazby mezi vstupy a výstupy:

- Vnější – popisuje relace „vstup - výstup“

- Vnitřní – popisuje relace „vstup – stav - výstup“

V případě lineárních a časově nezávislých systémů se ve funkci vnějších modelů spojitých soustav používají obrazové přenosy. Mimo přenosů se jako vnější modely používají frekvenční, případně přechodové a impulsní charakteristiky.

Z hlediska způsobu identifikace dělíme modely na:

- Analytické – matematický model sestavujeme na základě matematicko-fyzikální analýzy objektu. Můžeme určit jak stavové rovnice dynamického systému tak i vnější popisy systému.
- Experimentální – využíváme údaje a informace získané v průběhu pozorování nebo experimentu. Získáme matematický model vyjadřující vnější popis systému.

S ohledem na chování procesu v objektu dělíme experimentální modely na:

- Deterministické – na vstup přivádíme definovaný testovací signál. V objektech nepůsobí poruchové veličiny, nebo je lze zanedbat.
- Stochastické – projevují se v nich i náhodné vlivy (poruchy)

Dále se budu zabývat, s ohledem na Diplomovou práci, jen Experimentálními metodami identifikace.

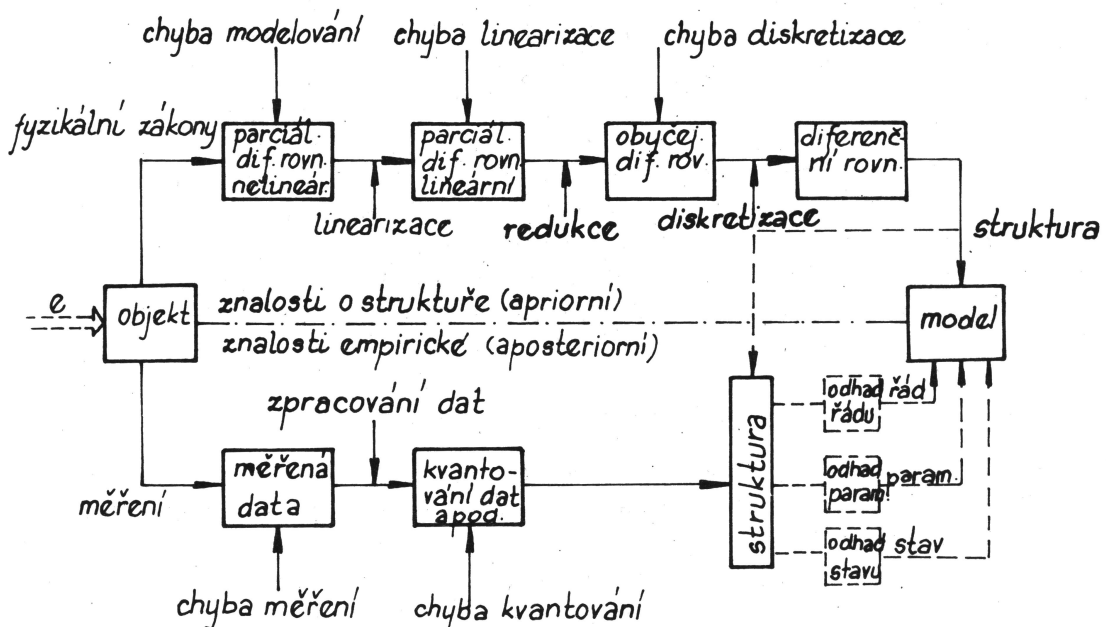
1.2 Experimentální metody identifikace

Při experimentální identifikaci využíváme údaje získané v průběhu pozorování nebo experimentu. Použitím souboru vstupních a výstupních dat, určíme matematický model vyšetřovaného objektu.

Veličiny, které jsme schopni při identifikaci na objektu měřit, dělíme na vstupní veličiny u (akční veličiny) a výstupy objektu y .

Pojmu experimentální identifikace dobře vyhovuje definice identifikace podle L. A. Zadeha: Identifikace je určení systému z dané třídy systému, ke kterému je testovaný systém na základě vstupů a výstupů ekvivalentní. V této definici testovaným systémem chápeme vyšetřovaný objekt a třídou systému třídu operátorů $S = \{O_m\}$ [1]

Při identifikaci používáme apriorní informace pro určení struktury modelu a měřená data pro získání nutných aposteriorních informací za účelem stanovení ekvivalence modelu s objektem. Tento postup je znázorněn ve schématu podle P. Eykhoffa na obrázku (Obr.1.)



Obr. 1. Schéma procesu identifikace

Experimentální metody identifikace můžeme rozdělit podle druhu použitých vstupních testovacích signálů.

Vstupní signály dělíme na:

- Přirozené – tj. provozní vstupní signály pozorované během provozu objektu (pasivní experiment)
- Umělé – uměle vytvářené vstupní signály s určitými vlastnostmi (aktivní experiment)

Signály dále dělíme na:

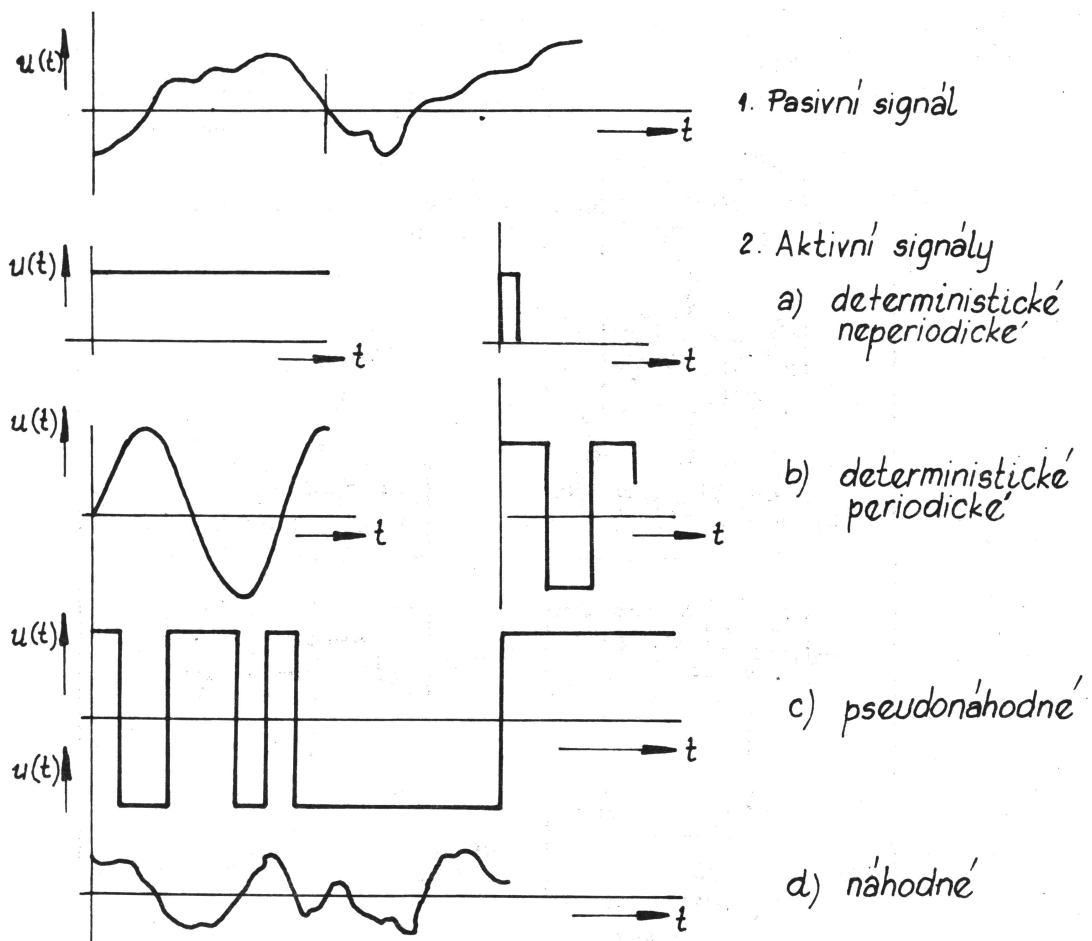
- Deterministické – průběhy těchto signálů jsou známými funkcemi času.
- Náhodné (stochastické) – jejich průběhy jsou neznámé nebo jsou to náhodné funkce času.

- Pseudonáhodné – jejich průběh je známý, v rámci jedné periody mají charakter známé realizace náhodného procesu a tyto realizace se periodicky opakují.

Deterministické vstupní signály dále dělíme na:

- Aperiodické – např. jednotkový skok
- Periodické – např. harmonický průběh, pravoúhlý průběh.

Příklady testovacích signálů jsou na obrázku (Obr. 2.)



Obr. 2. Příklady testovacích signálů

V Diplomové práci je použita Deterministická metoda identifikace.

1.3 Vyhodnocování přechodových charakteristik

Tato metoda patří mezi deterministické experimentální identifikace. Slouží pro parametrizaci modelů ze získaných neparametrických forem ve tvaru grafických průběhů nebo tabulačního zápisu výsledků získaných měření.

Využívají tzv. standardní testovací signály, mezi které řadíme:

- Jednotkový skok
- Jednotkový (Diracův) impuls
- Harmonický signál
- Obecný signál

Postup můžeme rozdělit do tří etap:

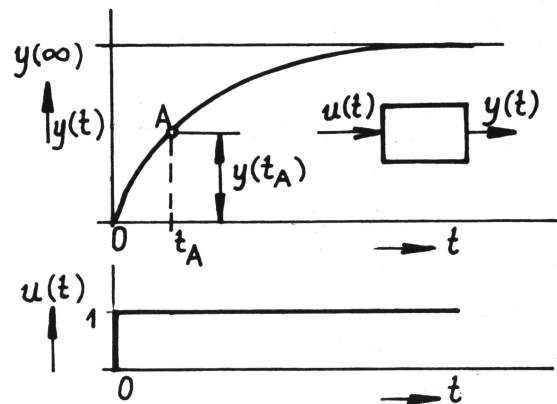
- Měření vstupně-výstupních závislostí.
- Určení neparametrického modelu vyhodnocením měření.
- Parametrizace získaného neparametrického modelu.

Přechodová charakteristika se měří tak, že se objekt nechá ustálit a potom vstupní veličinu skokem změním na jinou hodnotu. Časový průběh výstupní veličiny přepočítaný na jednotkovou změnu vstupní veličiny je přechodová charakteristika.

Přesné určení dynamických vlastností podle průběhu přechodových charakteristik je dost komplikované. Proto vyhodnocováním přechodových charakteristik se určuje většinou přibližný (aproximační) přenos objektu.

1.3.1 Aproximace soustavou prvního řádu bez dopravního zpoždění

Soustavou prvního řádu bez dopravního zpoždění můžeme aproximovat s dostatečnou přesností jen takové přechodové charakteristiky, které mají tzv. „prodlevu“ v okolí $t=0$ velmi malou. Viz. obrázek (Obr.3)



Obr. 3. Přejchodová charakteristika soustavy prvního řádu

Takovou soustavu můžeme popsat diferenciální rovnicí

$$a_1 y'(t) + a_0 y(t) = b_0 u(t) \quad (1)$$

kterou můžeme upravit

$$T y'(t) + y(t) = K u(t) \quad (2)$$

kde:

- $T = \frac{a_1}{a_0}$... časová konstanta soustavy
- $K = \frac{b_0}{a_0}$... zesílení soustavy

Rovnici (2) převedeme na přenos systému

$$G(s) = \frac{K}{Ts + 1} \quad (3)$$

a přechodovou funkci

$$h(t) = K \left(1 - e^{-\frac{t}{T}} \right) \quad (4)$$

V rovnicích (2) – (4) vystupují neznámé K a T . Zesílení K určíme z podmínky pro statickou rovnováhu soustavy, kterou lze formulovat jako podíl ustálené hodnoty výstupní veličiny k ustálené hodnotě vstupní veličiny.

$$K = \frac{y(\infty) - y(0)}{\Delta u(t)} = \frac{\Delta y(t)}{\Delta u(t)} \quad (5)$$

Časovou konstantu T určíme tak, že na experimentálně získané charakteristice zvolíme bod A se souřadnicemi t_A, y_A . Z rovnice (4) (pro $\Delta u(t) \neq 1$) pak platí:

$$y_A = \Delta y(t) \left(1 - e^{-\frac{t_A}{T}} \right) \quad (6)$$

Z rovnice (6) vypočítáme časovou konstantu T

$$T = -\frac{t_A}{\ln\left(1 - \frac{y_A}{\Delta y(t)}\right)} \quad (7)$$

1.3.2 Aproximace statických soustav vyšších řádů

V případě soustav vyšších řádů je aproximace jejich přechodových odezev dost náročný úkol. Když se přechodová charakteristika ustálí na konečné hodnotě, tak se jedná o statickou soustavu. Z tvaru přechodové charakteristiky statické soustavy vyššího řádu nelze přesně určit řád ani parametry soustavy. Proto používáme přibližné metody, pomocí kterých se určují pouze aproximační přenosy soustavy. Tyto metody se liší volbou aproximačního přenosu.

Jednu z nejjednodušších a snadno použitelných metod aproximace navrhl V. Strejc. Je vhodná pro objekty, které můžeme považovat za statické soustavy, složené ze sériově spojených jednodukapacitních článků. Předpokládáme, že časové konstanty jednotlivých článků jsou přibližně stejně velké. Tyto soustavy aproximujeme soustavami buď n -tého řádu se stejnými časovými konstantami nebo soustavami druhého řádu s různě velkými časovými konstantami.

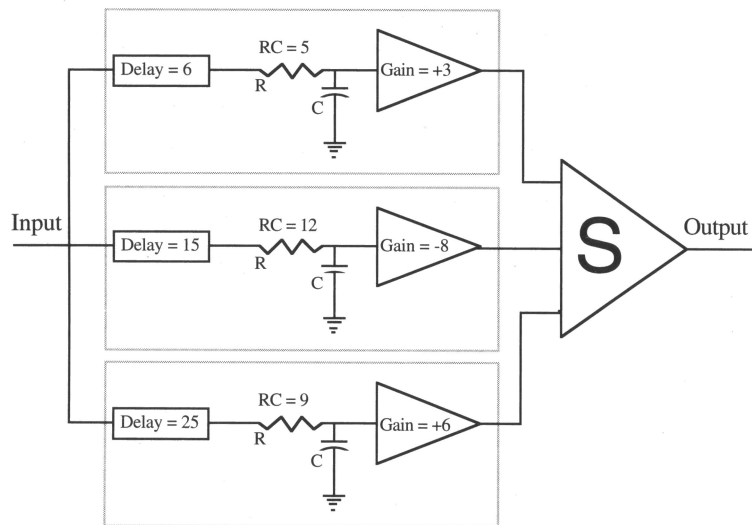
Postup aproximace viz. [1]

2 ŘEŠENÉ PŘÍKLADY IDENTIFIKACE

Příklad použití evolučního algoritmu a algoritmu symbolické regrese pro identifikaci je v [2]. V této kapitole si stručně představíme tento příklad.

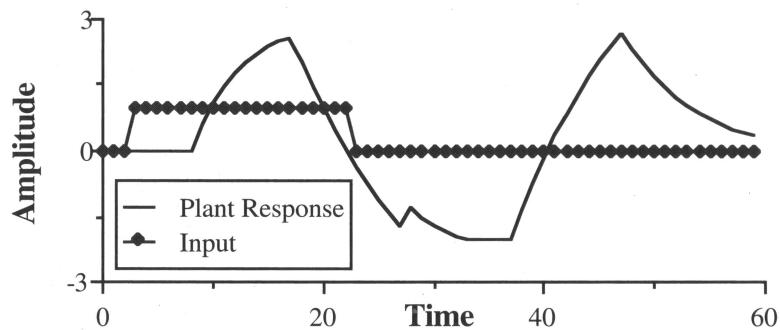
Pro mnoho problémů v oblasti regulace a řízení potřebujeme najít funkce, jako impulsní funkci nebo přechodovou funkci, pro systém, který nemá analytický model. Pro identifikaci je použito genetické programování [2,3]. Program je realizován v programovacím jazyce LISP.

Hledáme impulsní funkci lineárního t-invariantního systému. Jeho struktura je na obrázku (Obr. 4.)



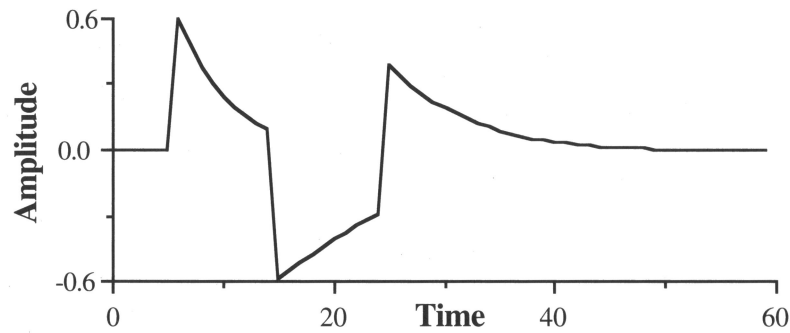
Obr. 4. Lineární t-invariantní systém

Na obrázku (Obr. 5.) je odezva systému na čtvercovou funkci jako akční veličinou. Čtvercová funkce v čase 3 přejde z 0 na 1 a v čase 23 přejde zpět na 0.



Obr. 5. Odezva systému na čtvercovou funkci

A na obrázku (Obr. 6.) je Impulsní charakteristika.



Obr. 6. Impulsní charakteristika systému.

Tento systém má impulsní funkci

$$\begin{cases} 0 & t < 6 \\ \frac{3\left(1-\frac{1}{5}\right)^{t-6}}{5} & \text{jinak} \end{cases} + \begin{cases} 0 & t < 15 \\ \frac{-8\left(1-\frac{1}{12}\right)^{t-15}}{12} & \text{jinak} \end{cases} + \begin{cases} 0 & t < 25 \\ \frac{6\left(1-\frac{1}{9}\right)^{t-25}}{9} & \text{jinak} \end{cases} \quad (8)$$

Výstup lineárního spojitého systému je dán rovnicí

$$o(t) = \int_{-\infty}^t i(t-t)H(t)dt \quad (9)$$

a výstup lineárního diskrétního systému je dán rovnicí

$$o(t) = \sum_{t=-\infty}^t i(t-t)H(t) \quad (10)$$

kde $i(t)$ je vstup (akční veličina) a $H(t)$ je impulsní funkce.

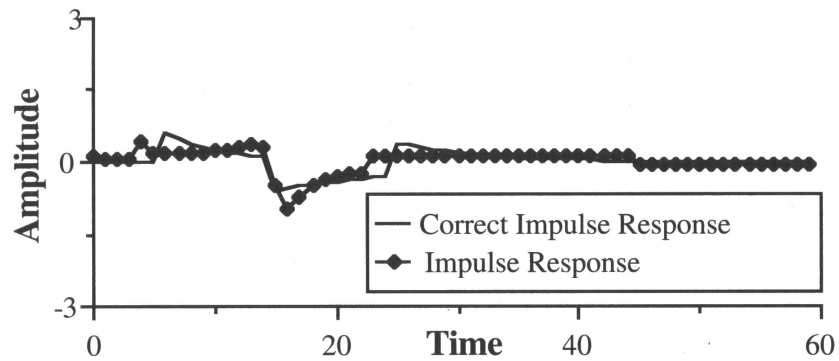
Množina funkcí použitých pro genetické programování je $F=\{+,-,*,\%,EXPP,IFLTE\}$, kde $\%$ je děleno, EXPP je funkce e^x a IFLTE je „If Less Then or Equal“.

Fitness se vypočítá z rovnice

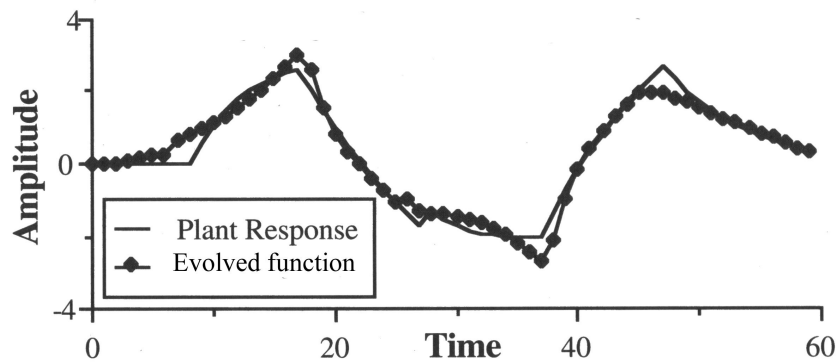
$$f(G) = \sum_{i=1}^{N_{fc}} \left[o(t_i) - \sum_{t=-\infty}^t i(t_i-t)G(t) \right]^2 \quad (11)$$

kde $N_{fc}=60$, $o(t)$ je výstup, $i(t)$ je vstup (akční zásah) a $G(t)$ je impulsní funkce.

Evolucí jsme nechali běžet 50 generací a potom jsme porovnali nejlepšího jedince z té generace s originální impulsní funkcí. Porovnání je na obrázku (Obr. 7.) a má Fitness 5,22, tj. 0,87 pro každý z 60 porovnávaných časových úseků. Na obrázku (Obr. 8.) porovnáваме vytvořenou funkci s odezvou systému na čtvercovou funkci.

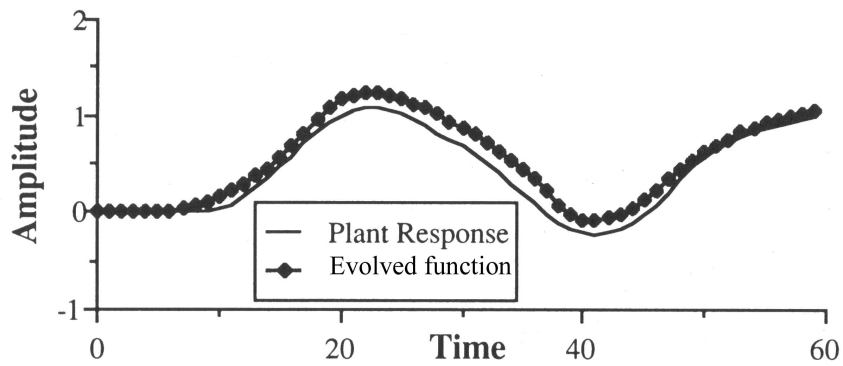


Obr. 7. Porovnání nejlepšího jedince s originální impulsní funkcí

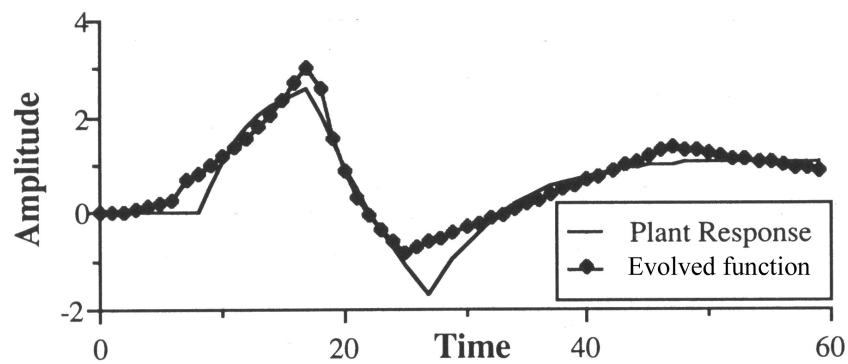


Obr. 8. Porovnání nejlepšího jedince s odezvou na čtvercovou funkci.

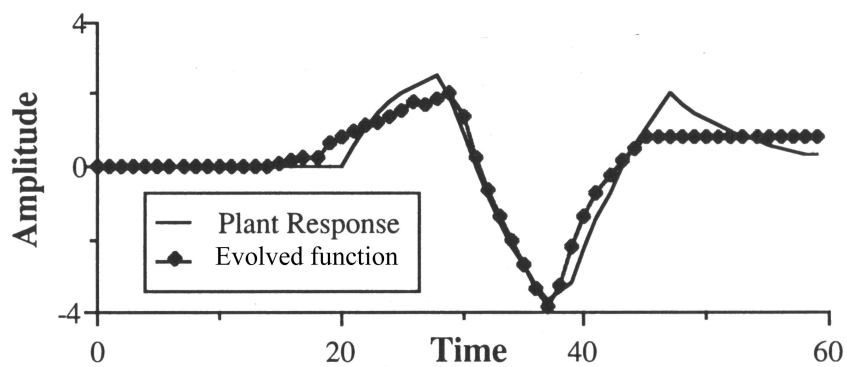
V regulaci se často objevují za akční veličiny funkce jako rampa nebo skok. Potom provedení evolučně vyvíjené impulsní funkce můžeme demonstrovat uvažováním dalších vstupních funkcí. A to: rampou (Obr. 9.), jednotkovým skokem (Obr. 10.), a krátkou čtvercovou funkcí (Obr. 11.).



Obr. 9. Porovnání nejlepšího jedince s odezvou na rampu.



Obr. 10. Porovnání nejlepšího jedince s odezvou na jednotkový skok.



Obr. 11. Porovnání nejlepšího jedince s odezvou na krátkou čtvercovou funkci.

3 EVOLUČNÍ ALGORITMY

Evoluční algoritmy náleží do oblasti umělé inteligence, která se nazývá softcomputing. Tyto algoritmy se používají pro nalezení nejlepšího řešení (výsledku) v optimalizačních problémech. Většina problémů v inženýrské praxi může být definována jako optimalizační problém jako např. nalezení optimální cesty robota, optimální tloušťky stěny tlakové nádoby, optimální nastavení parametrů regulátoru, optimální relace mezi fuzzy množinami a ve spojení s vhodnou metodou symbolické regrese (např. analytické programování) identifikace systému, atd.

Optimalizační problém je definice problému takovým způsobem, že nejlepší řešení leží v globálním extrému. Tím bývá obvykle globální minimum. Zde používáme výrazy účelová funkce (cost function), hodnota účelové funkce (cost value, fitness) a parametry. Účelová funkce je analytické vyjádření optimalizovaného problému a hodnota účelové funkce je hodnota, kterou dostaneme po dosazení parametrů do účelové funkce.

V následujících podkapitolách stručně popíšeme vybrané evoluční algoritmy a SOMA algoritmus použitý pro vypracování řešení pomocí analytického programování.

3.1 GA – Genetický algoritmus

Genetický algoritmus je skupina metod, které se dají použít k řešení optimalizačních problémů. První Genetický algoritmus představil John H. Holland v roce 1975. Genetický algoritmus je založen na principech biologické evoluce, jak je popsal Charles Darwin. Mnoho termínů z biologické evoluce se též používají i v Genetickém algoritmu. Genetický algoritmus pracuje s populací jedinců. Parametry jsou vyjádřené v binárním kódu. Pro činnost Genetického algoritmu potřebujeme mít definovanou účelovou funkci a fitness, které říkají, jaké řešení je dobré a jaké je špatné. Podle této funkce se vybírají rodiče.

3.1.1 Kódování a fitness funkce

V předchozím textu bylo řečeno, že každý jedinec má parametry nazvané geny, které jsou kódované v binární formě. Všechny geny jednoho jedince vytvářejí řetězec nazývaný chromozóm. Sada parametrů reprezentovaná konkrétním chromozómem se nazývá genotyp a hodnota účelové funkce chromozómu se nazývá fenotyp [4]. Hodnota účelové funkce říká jak je úspěšné řešení. Další termín spojený s Genetickým algoritmem je fitness funkce

a hodnota fitness funkce. Toto se používá k výběru rodičů, protože říká, jak jsou jednotlivá řešení vhodná.

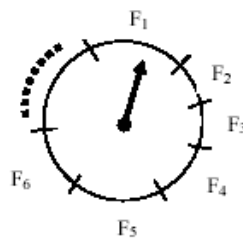
Fitness je v [4] definován jako:

$$F(\mathbf{a}) = \frac{1}{f_{\min} - f_{\max}} [(1 - \epsilon)f(\mathbf{a}) + f_{\min} \epsilon - f_{\max}] \quad (12)$$

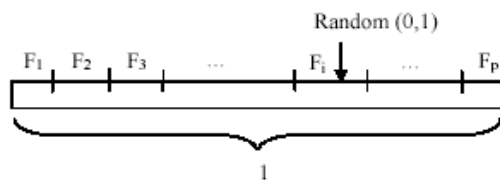
kde:

- $F(\alpha)$... fitness pro jedince α
- f_{\min} ... minimální hodnota účelové funkce
- f_{\max} ... maximální hodnota účelové funkce
- ϵ ... malé kladné číslo blízké nule (např. 0,001)

Rovnice (4) přiřazuje fitness každému jedinci v populaci v intervalu $\langle 0, 1 \rangle$ podle jeho hodnoty účelové funkce. Jedinec s minimální hodnotou účelové (nejlepší) funkce má fitness 1 a jedinec s největší hodnotou účelové funkce (nejhorší) má fitness 0. Fitness reprezentuje interval na jednotkovém kruhu (Obr.12.) nebo na jednotkové úsečce (Obr.13.)



Obr.12. Schéma ruletového kola na jednotkovém kruhu

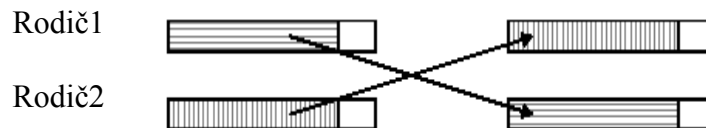


Obr. 13. Schéma jednotkové úsečky

3.1.2 Reprodukce

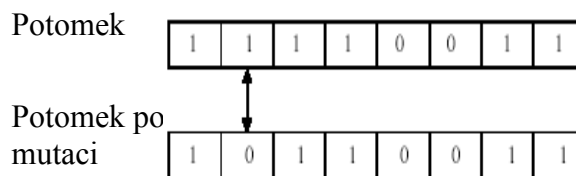
Reprodukce vyžaduje nejdříve vybrání vhodných rodičů. To můžeme udělat kvazi náhodně pomocí Ruletového kola na jednotkovém kruhu. Jedinec s lepší hodnotou účelové funkce bude pravděpodobněji vybrán než jedinec s velkou hodnotou účelové funkce. Je tu možnost opětovného vybrání. Vybraní rodiče se přemístí do „pářícího jezírka“. Zde jsou 2 rodiče náhodně vybráni aby vytvořili 2 potomky. Jejich chromozómy se rekombinují ve smyslu křížení a mutací.

Křížením se myslí rozdělení chromozómů v náhodně vybraném místě a vyměnění částí mezi rodiči. Viz obrázek (Obr. 14.)



Obr. 14. Schéma křížení

Po křížení každý potomek prochází procesem mutace. Jeden bit mutuje, například, když má bit hodnotu 0 je změněn na 1 a naopak. Viz obrázek (Obr. 15.)



Obr. 15. Schéma mutace

3.2 SA – Simulované žihání

Simulované žihání představil počátkem 80. let 20. století Kirkpatrick. Inspirací pro vývoj tohoto algoritmu bylo žihání kovu. Zde je kov zahřán až skoro k bodu tání a potom velmi pomalu ochlazován. Záměrem je eliminovat nestabilní částice. Jinými slovy, částice jsou posouvány do optimálního energetického stavu. Kov má pak stabilnější a stejnoměrnější krystalickou strukturu.

Tenhle přístup byl použit v případě simulovaného žihání včetně terminologie. Simulované žihání je lepší varianta Horolezeckého algoritmu [4]. Oba algoritmy startují z náhodně zvoleného bodu. Pak se v sousedství generuje určitý počet bodu (záleží na uživateli).

V případě Horolezeckého algoritmu, bod s nejlepší hodnotou účelové funkce je zvolen za střed nového sousedství (nový startovací bod). Pokud je nejlepší hodnota účelové funkce ve startovacím bodě, pak je tento bod zvolen pro další běh. Tato subrutina se opakuje několikrát (znovu záleží na uživateli). Horolezecký algoritmus ve většině případů nachází lokální extrém. Simulované žíhání má toto ošetřené pomocí principu akceptování řešení.

Princip akceptování řešení během běhu Simulovaného žíhání je následující. Když nová hodnota účelové funkce je lepší než stará, je hned přijata. To znamená, že rozdíl mezi těmito dvěma hodnotami je záporný. Jestliže je rozdíl kladný, tj. nová hodnota je horší než stará, generuje se číslo z intervalu $\langle 0,1 \rangle$. Jestli toto číslo je menší než pravděpodobnost vypočtena z rovnice (13), je tento nový bod přijmut. V opačném případě starý bod pokračuje do dalšího zpracování. Tento postup se jmenuje Metropolisovo kritérium.

$$p(t) = e^{-\frac{\Delta E}{T}} \quad (13)$$

kde:

- $p(t)$...pravděpodobnost přenosu pro teplotu
- ΔE ...rozdíl mezi hodnotami účelové funkce starého a nového bodu
- T ...současná hodnota teploty – řízený parametr pro ochlazování

Algoritmus začíná s nejvyšší teplotou, která se po krocích snižuje. Rovnice (14) ukazuje standardní ochlazovací funkci.

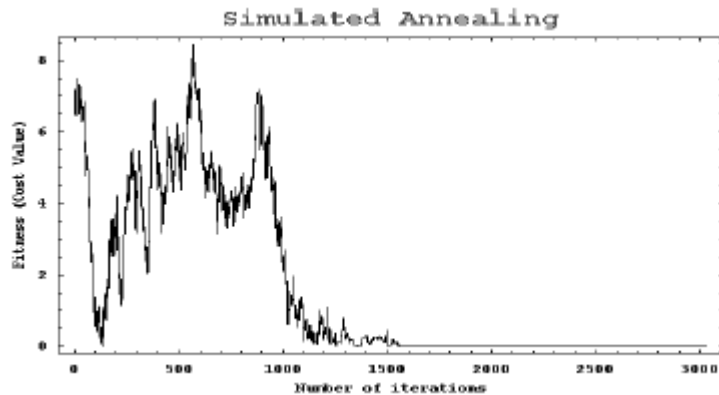
$$T_{n+1} = \alpha T_n \quad (14)$$

kde:

- T_{n+1} ...teplota v dalším kroku
- T_n ...současná teplota
- α ...chladicí koeficient z intervalu $\langle 0,1 \rangle$

Jestliže je α blízká 0, řešení konverguje do lokálního extrému. Na druhé straně, když α je blízká 1, chlazení je velmi pomalé a trvá dlouho, než se najde optimum.

Na obrázku (Obr. 16.) je graf znázorňující typický průběh Simulovaného žíhání.



Obr. 16. Typický průběh Simulovaného žíhání

3.3 DE – Diferenciální evoluce

Diferenciální evoluce je známá od roku 1995. Jejími otci jsou Ken Price a Rainer Storn.

Algoritmus pracuje s populací jedinců, ale je zde jedna odlišnost oproti ostatním evolučním algoritmům. K vytvoření potom jsou potřeba místo dvou rodičů, čtyři. Princip Diferenciální evoluce je na obrázku (Obr. 17.) [5]

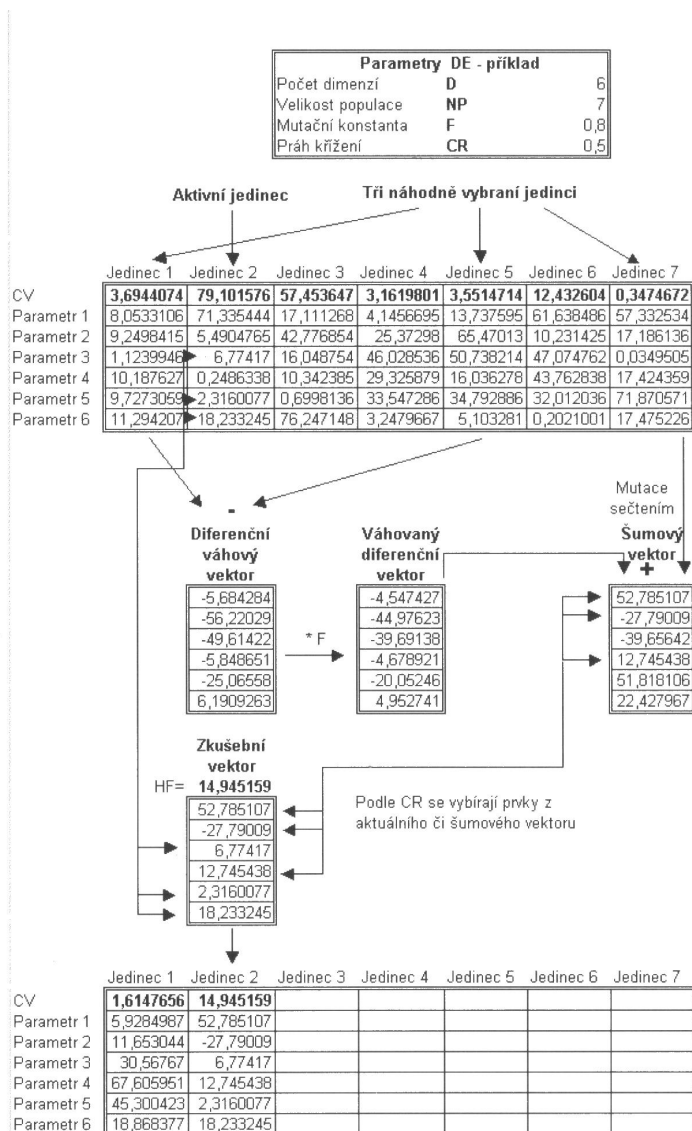
3 jedinci jsou náhodně vybraní k aktivnímu jedinci. Pak jeden z náhodně vybraných rodičů je odečten od druhého rodiče a tím se vytvoří diferenční vektor. Tento se vynásobí s mutační konstantou F a vznikne váhový diferenční vektor. Třetí rodič plus váhový diferenční vektor dá šumový vektor. Potom vytvoříme zkušební vektor, kde vystupují některé argumenty z aktivního jedince a některé ze šumového vektoru. To záleží na hodnotě prahu křížení CR . Viz tabulka (Tab. 1.) kde jsou prezentovány řídicí a ukončovací parametry DE [1]

Tab. 1. Parametry DE

Jméno parametru	Doporučený rozsah	Poznámka
F	<0,2>	Řídicí parametr
CR	<0,1>	Řídicí parametr
Dim	Dáno problémem	
PopSize	<10, na uživateli>	Řídicí parametr
Generation	<na uživateli>	Ukončovací parametr

- F...mutační konstanta, číslo, kterým se násobí diferenční vektor k získání váhového diferenčního vektoru. Doporučená hodnota je mezi 0,3 a 0,9.
- CR...práh křížení, říká jestli argumenty aktivního jedince nebo zkušebního vektoru se použijí v novém jedinci.
- Dim...dimenze problému, Jde o počet argumentů účelové funkce. Parametr je dán řešeným problémem a lze ho změnit pouze reformulací problému.
- PopSize...velikost populace.

Existuje 10 variant Diferenciální evoluce [5]. Rozdíl mezi nimi spočívá v používání šumového vektoru.



Obr. 17. Princip Diferenciální evoluce

3.4 SOMA – Samo-organizující se migrační algoritmus

SOMA existuje od roku 1999. Otcem tohoto algoritmu je Doc. Ivan Zelinka, PhD. SOMA pracuje se skupinou jedinců (populací), jejíž chování můžeme popsat jako Strategie soutěže a spolupráce. Vytvoření nové populace jedinců není založené na evolučních principech (2 rodiče na jednoho potomka), ale na chování sociální skupiny. Během jedné generace, v případě SOMy nazývané Migrace, se mění na hledané ploše pouze pozice jedince [5].

Princip SOMA algoritmu je na obrázku (Obr. 18.). V každé migraci se najde nejlepší jedinec, který má nejmenší hodnotu účelové funkce, kterého nazýváme Leader. Aktivní jedinec z populace se pohybuje ve směru k Leadrovi. Po ukončení pohybu najdeme jedince s minimální hodnotou účelové funkce. Jestliže je lepší než hodnota jedince ze staré populace, bereme tohoto jedince na nové pozici do další generace . Jinak starý jedince zůstává. Pohyb je vysvětlen rovnicí (15).

$$x_{i,j}^{MK+1} = x_{i,j,START}^{MK} + (x_{L,j}^{MK} - x_{i,j,START}^{MK}) * t * PRTVector_j \quad (15)$$

kde:

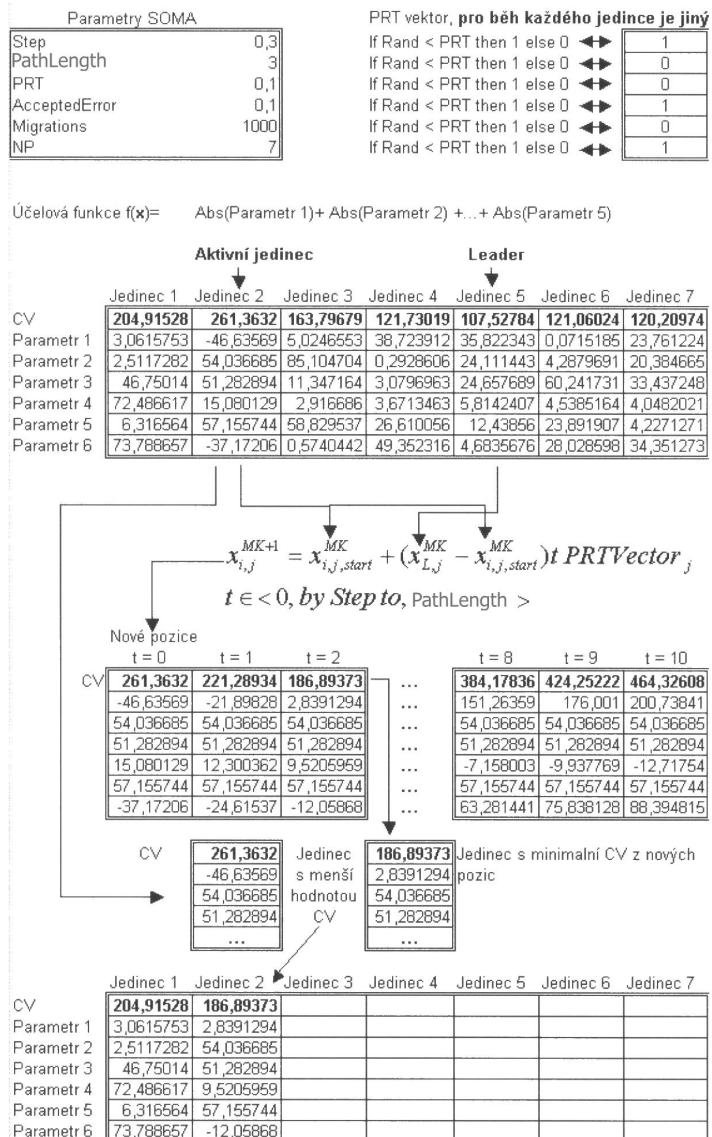
- $x_{i,j}^{MK+1}$...hodnota j-tého parametru i-tého jednice v dalším migračním kole MK+1.
- $x_{i,j,START}^{MK}$...hodnota j-tého parametru i-tého jednice, Startovací pozice v současném migračním kole.
- $x_{L,j}^{MK}$...hodnota j-tého parametru Leadra v migračním kole MK
- t ...krok $\in \langle 0, step, PathLength \rangle$
- PRTVector...vektor jedniček a nul závisící na PRT. Jestliže náhodné číslo z intervalu $\langle 0, 1 \rangle$ je menší než PRT, pak do PRTVectoru uložíme 1, a naopak.

SOMA pracuje s řídicími a ukončovacími parametry, které jsou shrnuty v tabulce (Tab. 2.)

Tab. 2. Parametry SOMA

Parametr	Doporučený rozsah	Poznámka
PathLength	<1,1; 3>	Řídicí parametr
Step	<0,11; PathLength>	Řídicí parametr
PRT	<0; 1>	Řídicí parametr
Dim	Dáno problémem	
PopSize (NP)	<10, na uživateli>	Řídicí parametr
Migrations	<10, na uživateli>	Ukončovací parametr
MinDiv	< jakýkoliv, na uživateli >	Ukončovací parametr

- PathLength...tento parametr určuje, jak daleko se aktivní jedinec zastaví od Leadra. Při PathLength=1 se aktivní jedinec zastaví na pozici Leadra a při PathLength=2 za ním ve stejné vzdálenosti z jaké startoval, atd.
- Step...Určuje krok z jakým se bude mapovat cesta aktivního jedince.
- PRT...perturbace, Podle tohoto řídicího parametru se určuje pertubační vektor PRTVector, který ovlivňuje to,zda se aktivní jedinec bude pohybovat přímo k Leadrovi či ne.
- Dim...udává počet argumentů účelové funkce. Je dán samotným problémem a může být změněn pouze předefinováním celého problému.
- PopSize(NP)...velikost populace.
- Migrations...je ekvivalentem parametru generace z jiných evolučních algoritmů. Udává , kolikrát se populace přeorganizuje.
- MinDiv...definuje maximální rozdíl mezi nejhorším a nejlepším jedincem v aktuální populaci. Jestliže je rozdíl menší než MinDiv pak je běh algoritmu ukončen.



Obr. 18. Princip SOMA

Jsou 4 verze SOMA algoritmu [5]:

- AllToOne
- AllToAll
- AllToAdaptive
- AllToOneRand

AllToOne – verze popsaná v předchozím textu. Je zde Leader a ostatní jedinci migrují k němu.

AllToAll - zde není žádný Leader. Každý jedinec migruje k ostatním jedincům. Jedinec se přesune do nejlepší pozice ze všech jeho NP-1 cest po všech migracích v jednom kole.

AllToAllAdaptive – tento algoritmus je podobný jako AllToAll, s tím rozdílem, že migrující jedinec se nepřesouvá do nové pozice až po všech migracích ke všem ostatním, ale po každé, aktuálně dokončené migraci ke každému z NP-1 jedinců se přesune na lepší pozici nalezenou na této aktuální migraci. Z této pozice pak provádí migraci k dalším zbývajícím jedincům.

AllToOneRand – podobně jako v AllToOne se jedinci pohybují k Leadrovi, který je vybrán náhodně.

V diplomové práci používám SOMA algoritmus v modifikaci AllToOne.

3.5 Počet vyhodnocení účelové funkce pro jednotlivé algoritmy

Někdy je dobré vědět jak dlouho bude evoluce trvat. Abychom toto zjistili, potřebujeme znát kolikrát se ohodnotí účelová funkce. Pak toto číslo vynásobíme časem potřebným k jednomu ohodnocení účelové funkce.

Výpočet pro ohodnocení účelové funkce pro SOMA algoritmus verzi AllToOne je rovnice (16), pro verzi AllToAll je to rovnice (17). Pro DE a GA je to rovnice (18) a pro SA rovnice (19).

$$N_{eval} = \frac{(NP-1) * PathLength * Migrations}{step} \quad (16)$$

$$N_{eval} = \frac{NP * (NP-1) * PathLength * Migrations}{step} \quad (17)$$

$$N_{eval} = NP * Generations \quad (18)$$

$$N_{eval} = TempIter * MaxIter \quad (19)$$

kde:

- N_{eval} ...počet ohodnocení účelové funkce.
- NP...velikost populace, počet jedinců.
- PathLength...řídící parametr SOMA, viz tabulka (Tab.2).

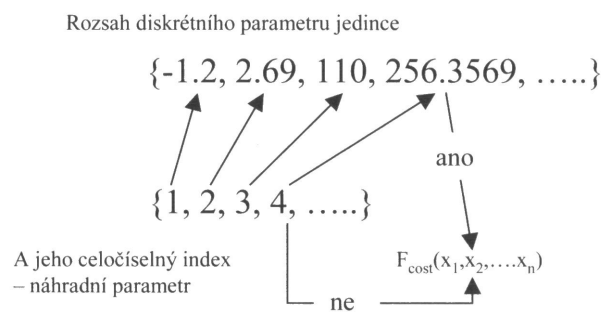
-
- Migrations...ukončovací parametr SOMA, počet migračních kol.
 - Step...řídící parametr SOMA.
 - Generations...parametr pro DE a GA, počet generací.
 - TempIter...počet iterací pro každou teplotu
 - MaxIter...počet sestupných kroků teploty.

4 AP - ANALYTICKÉ PROGRAMOVÁNÍ

Autorem tohoto algoritmu je Doc. Ivan Zelinka. Na rozdíl od jiných algoritmů symbolické regrese není jeho běh vázán na konkrétní evoluční algoritmus, nebo programovací jazyk. V diplomové práci pracuje se SOMA evolučním algoritmem. Spojení se SOMA programováním je stejné jako u množiny diskretních hodnot [5, 6, 7].

4.1 Množina diskretních hodnot

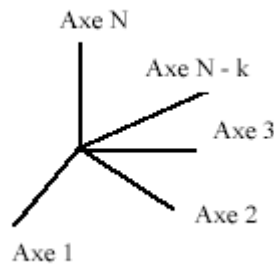
Evoluční algoritmy pracují se Specimenem [5], kde jsou definovány dovolené intervaly argumentů účelové funkce. Počáteční populace se náhodně vybere podle těchto limitů. Někdy požadujeme pouze diskretní hodnoty. Pak populace pracuje s indexy. Ty obsahují množinu diskretních hodnot (např. $-1,156; 2; 6,5$) a vytvoří se celočíselný index pro každé číslo této množiny (např. $1,2,3$). Množina indexů jsou celá čísla od 1 do velikosti množiny diskretních hodnot. Během vzniku populace, se náhodně berou hodnoty z celočíselného indexu. Index je nahrazen odpovídající diskretní hodnotou před vyhodnocením účelové funkce. Viz obrázek (Obr. 19.)



Obr. 19. Princip ošetření diskretního parametru jedince

4.2 Obecný funkční prostor

Analytické programování pracuje s komplikovaným funkčním prostorem, kde není požadavek na ortogonalitu, ortonormalitu a linearitu. Podle těchto vlastností to nazýváme Obecný Funkční prostor (General Function Space –GFS). Každý bod na osách GFS reprezentuje funkci, operátor nebo konstantu. Viz obrázky (Obr. 20.) a (Obr. 22.)



Obr.20.Obecný funkční prostor

4.3 Základní principy

GFS obsahuje funkce a jejich argumenty, operátory, proměnné, konstanty, atd. Množina funkcí AP (GFS) je rozdělena na několik podmnožin podle počtu argumentů. Na obrázku (Obr. 21.) je rozdělení množin GFS, kde GFS_{0arg} jsou funkce bez argumentu (např. konstanty, proměnné). GFS_{All} je množina všech funkcí ze všech ostatních podmnožin.

$$\begin{aligned}
 GFS_{all} &= \{+, -, /, ^, d/dt, Sin, Cos, Tan, t, C1, Mod, \dots\} \\
 GFS_{3arg} &= \{BetaRegularized, \dots\} \\
 GFS_{2arg} &= \{+, -, /, ^, Log, Mod, GammaRegularized\dots\} \\
 GFS_{1arg} &= \{Sin, Cos, Tan, Abs, Re, Im, \dots\} \\
 GFS_{0arg} &= \{t, x, y, z, C1, C2, Kinchin, \dots\}
 \end{aligned}$$

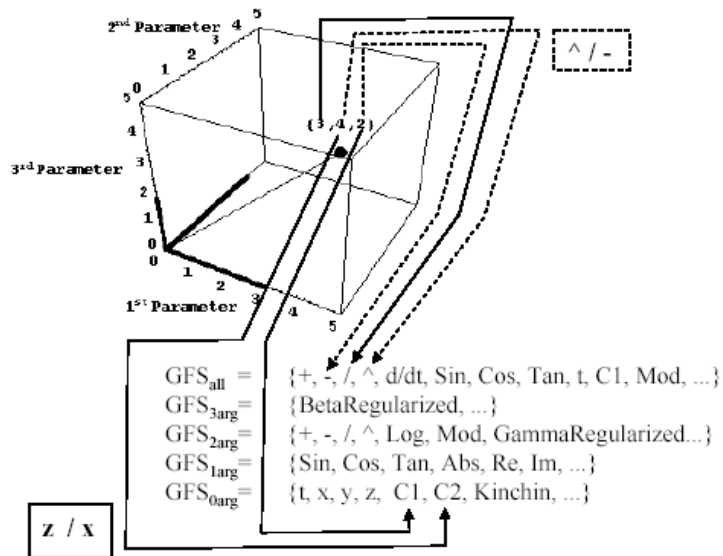
Obr. 21. Rozdělení funkcí podle počtu argumentů

Na obrázku (Obr. 22.) je příklad tvoření funkce z GFS. V tomto případě je znázorněné jak je AP ošetřeno proti vzniku patologických funkcí.

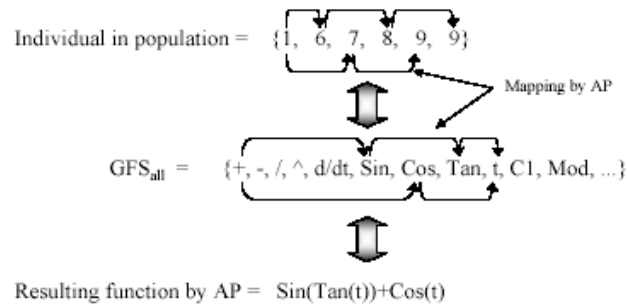
Postup je následující: První index je nahrazen odpovídající funkcí z GFS_{All} . Pak subrutina v AP zkontroluje kolik argumentů vybraná funkce potřebuje a kolik indexů jedince zbývá do konce. Ostatní funkce se vybírají na základě těchto faktů.

Na obrázku (Obr. 23.) je ukázka „mapování“ jedince bez možnosti vzniku patologické funkce a na obrázku (Obr. 24.) je ukázáno ošetření proti vzniku patologické funkce. [6, 7]

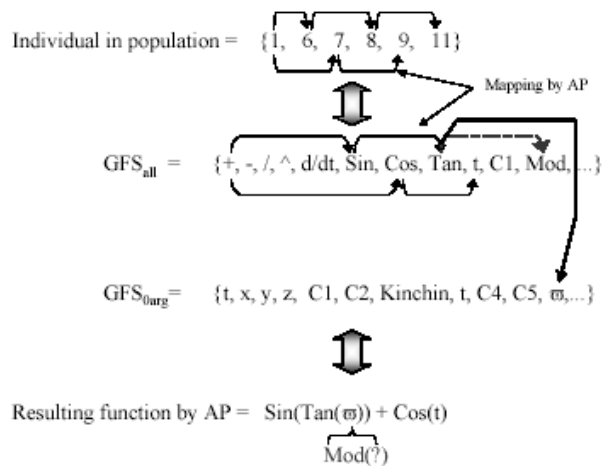
Když AP při „mapování“ jedince narazí na funkci, které potřebuje určitý počet argumentů, a již tento počet není k dispozici, tak nepřizpůsobí tomuto indexu funkci z GFS_{All} , ale přiřadí mu funkci z GFS_x , tak aby byly dodrženy počty argumentů. Tj.vytvoří se nepatologická funkce. Viz obrázky (Obr. 22.) a (Obr. 24.)



Obr. 22. Obecný funkční prostor GFS – příklad

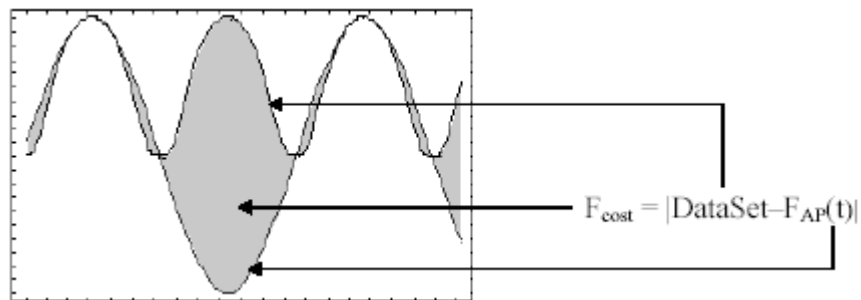


Obr. 23. Mapování jedince – nepatologický případ



Obr. 24. Mapování jedince – patologické případ

Analytické programování je použito k vytvoření nových funkcí. Např. funkcí které korespondují s neznámými daty. Vhodnou účelovou funkcí pro takovéto případy je absolutní hodnota z rozdílu mezi originálními daty a nalezenou funkcí, jak to popisuje obrázek (Obr. 25.). Cílem AP je minimalizovat rozdíl, tj. minimalizovat šedou plochu na obrázku (Obr. 25.). V nejlepším případě by měla být hodnota účelové funkce blízká nule.



Obr. 25. Účelová funkce v AP

V AP se nepočítá hodnota účelové funkce spojitě pro všechny body, ale interval je vzorkován a v takto získaných bodech se počítá rozdíl. V podstatě, účelová funkce je suma těchto rozdílů.

4.4 Nalezení výsledné funkce

V průběhu AP jsou funkce nalezeny s obecnými konstantami K , indexovanými $K[[1]]$, $K[[2]]$, ..., $K[[n]]$. V pozdějších provedeních AP tyto konstanty jsou nahrazeny konstantami $VarA$, $VarB$, ... kvůli lepší přehlednosti. Pro vyhodnocení účelové funkce je potřeba tyto obecné konstanty vyčíslit. Toto vyčíslení provádíme v Diplomové práci standardní metodou nelineární fitování.

4.5 Implementace AP

Pro implementaci AP a také evolučního algoritmu používáme software Mathematica firmy Wolfram (www.wolfram.com).

Tento program pracuje se soubory nazývanými notebooky. Notebooky pracují s buňkami, kde se zapisují příkazy. Funkce musí být před použitím v notebooku definovány, proto v notebooku jsou sekce s AP. První sekce se jmenuje FUNCTIONS, definuje funkce programu pro práci s jedinci a jejich parametry.

4.5.1 MainAP, množina funkcí

Příloha (P I) obsahuje kód funkce MainAP. V této funkci AP jsou definovány různé funkce a konstanty. Ty jsou rozděleny do různých skupin, jako např. Elementární funkce, Základní aritmetické funkce, atd. Každá funkce má následující strukturu: jméno, počet argumentů a interval na kterém chceme funkci používat.

Pak jsou všechny tyto skupiny sloučeny do jedné velké skupiny funkcí. Která se potom rozdělí na skupiny podle počtu argumentů. Ty jsou potom ještě rozděleny do skupiny TwoOneZeroFunction, kde jsou funkce s počtem argumentů 0,1,2, do skupiny OneZeroFunction, kde jsou funkce s počtem parametrů 0,1 a skupiny ZeroFunction, kde jsou konstanty a proměnné. Tyto skupiny funkcí slouží v AP ke generování nepatologických funkcí.

4.5.2 PickUp funkce

Tyto funkce jsou též zobrazeny v příloze (P I). Tyto funkce přiřazují indexům v jedinci funkce z množiny funkcí. Jsou to 4 funkce stejného jména, lišící se vstupními parametry. Pravidlo pro rozhodování, kterou funkci použít je založené na tom, kolik parametrů zbývá v jedinci do konce. Když je zde 3 a více parametrů tak se vybírá z množiny GFS_{All}. Když jsou zde jen 2 parametry tak se vybírá z množiny TwoOneZeroFunctions. Podobně pro 1 a 0 zbývajících parametrů.

4.5.3 GiveFunction

GiveFunction (příloha P I) používá PickUp funkci a dává všemu formu. Jsou zde také subrutiny, které indexují konstanty K. Jestliže se v jedinci objeví patologické chování, penalizuje ho funkce hodnotou účelové funkce 10^6 .

V této softwarové implementaci je ještě jedna penalizace. V nastavení CriticalParameterSet (Obr. 26.) se nastavuje časový limit pro výpočet účelové funkce.

```
CriticalParameterSet[]:=Module[{},  
    $MaxPrecision=10;  
    TimeLimit=10;  
    TimeLimitRetVal=1000000.;  
]
```

Obr. 26. CriticalParameterSet

4.5.4 CostValue

Tato funkce nachází konečnou podobu AP včetně přiřazení číselných hodnot konstantám. Účelová funkce se nepočítá ve všech bodech, ale jen v pár bodech. Podle [3] 50 bodů bylo zvoleno pro výpočty. Potom výsledná účelová funkce je suma rozdílů mezi originálními daty a nalezenou funkcí ve vzorkovacích bodech.

II. PRAKTICKÁ ČÁST

5 VYPRACOVÁNÍ V AP

Pro vypracování identifikace jsme si zvolili evoluční algoritmus SOMA a jeho variantu AllToOne. S parametry:

- PathLength = 3,0
- Step = 0,11
- PRT = 0,1
- NP = 200
- Migrations = 10
- MinDiv = -0,1

Funkce CostValue AP je na obrázku (Obr. 27.) a definice MainAP je na obrázku (Obr. 28.)

```

CostValue[fce_] := Module[{RetFce, cv1, cv2},
  If[fce[[2]] > 0,
    {
      tmp = fce;
      or = OutputResponse[TransferFunction[s, tmp[[1]]], UnitStep[t], t][[1]];
      CostFce[K_] := Evaluate[or];
      var = Take[char, fce[[2]]];
      RetFce = NonlinearFit[data, CostFce[var], {t}, var];
    },
    RetFce = fce[[1]];
  temp = Abs[Re[Table[outres - N[Re[RetFce]], {t, 0, 10, .1}]]] /.
    {Indeterminate -> 10000., ComplexInfinity -> 10000., Infinity -> 10000.};
  cv1 = If[ones.temp > 1000000, 1000000., ones.temp];
  cv2 = If[Head[cv1] == Real, 1.0 cv1, 1000000., 1000000.];
  If[cv2 < 1, Print[pocitadlo, "\n", cv2, "\n", fce, "\n", RetFce];
    {pocitadlo, cv2, fce, RetFce} >>> test3.txt; Abort[], Null];
  pocitadlo++;
  Return[cv2]
]

```

Obr. 27. CostValue funkce

```

BasicArithmetic = {
  {Plus, 2, {}}, {Subtract, 2, {}}, {Minus, 1, {}}, {Times, 2, {}}, {Divide, 2, {}}
};

ElementaryFunctions = {
  {Power, 2, {}}
};

VariousConstants = {{s, 0, {}}, {K, 0, {}}};

MainAP[BasicArithmetic, VariousConstants];

```

Obr. 28. Definice funkce MainAP

Pro simulace jsme zvolili tyto systémy:

$$\frac{1}{s^2 + 2s + 1} \quad (19)$$

$$\frac{1}{(s + 1)^3} \quad (20)$$

$$\frac{1 - s}{(1 + s)^3} \quad (21)$$

$$\frac{1 - 0,5s}{s(s + 1)} \quad (22)$$

$$\frac{1}{(1 + s)(1 + 0,5s)(1 + 0,5^2 s)(1 + 0,5^3 s)} \quad (23)$$

kde systémy (19),(20),(21),(23) jsou stabilní a systém (22) je systém nestabilní. Nestabilní systém jsme zvolili z důvodu ověření schopnosti identifikace i pro nestabilní systémy.

6 VÝSLEDKY SIMULACÍ

Pro každý systém jsme nechali přes 15x proběhnou evoluci. Kvůli vážným technickým problémům nebylo možné udělat simulace systému (21) a pro systém (23) se povedlo udělat jen 3 simulace. V tabulce (Tab. 3.) jsou vyjádřeny dílčí výsledky. A to počet simulací, minimální, maximální a průměrná hodnota účelové funkce, a minimální, maximální a průměrný počet ohodnocení účelové funkce potřebný k nalezení výsledku.

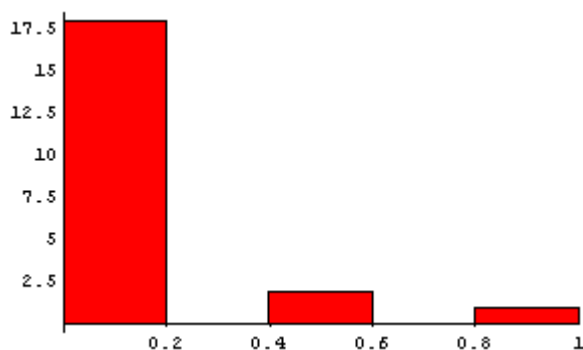
Tab. 3 Výsledky simulací statisticky

system	No	Min N _{Eval}	Max N _{Eval}	Ø N _{Eval}	Min CV	Max CV	Ø CV
$\frac{1}{s^2 + 2s + 1}$	21	31	3358	718,333	0	0,892707	0,0976925
$\frac{1}{(s + 1)^3}$	16	56	2432	1110,81	0,494889	0,935361	0,768286
$\frac{1 - 0,5s}{s(s + 1)}$	22	2	801	180,955	$3,1 \cdot 10^{-14}$	0,685508	0,113692
$\frac{1}{(1+s)(1+0,5s)(1+0,5^2s)(1+0,5^3s)}$	3	38	221	99,3333	0,37409	0,512158	0,420114

Kde:

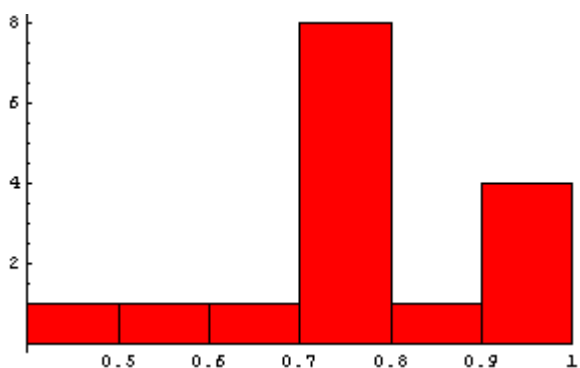
- No...počet simulací
- Min N_{Eval}...minimální počet ohodnocení účelové funkce potřebný k nalezení výsledku
- Max N_{Eval}... maximální počet ohodnocení účelové funkce potřebný k nalezení výsledku
- Ø N_{Eval}... průměrný počet ohodnocení účelové funkce potřebný k nalezení výsledku
- Min CV...minimální hodnota účelové funkce
- Max CV...maximální hodnota účelové funkce
- Ø CV...průměrná hodnota účelové funkce

Výsledky pro systém (19) jsou v příloze (P II) a grafický průběh v příloze (P VI). Histogram je na obrázku (Obr. 29.)



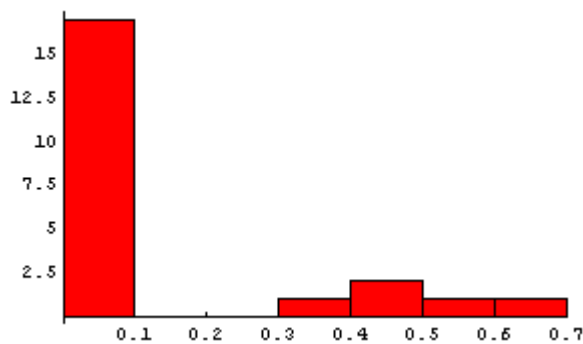
Obr. 29. Histogram pro systém (19)

Výsledky pro systém (20) jsou v příloze (P III) a grafický průběh v příloze (P VII). Histogram je na obrázku (Obr. 30.)



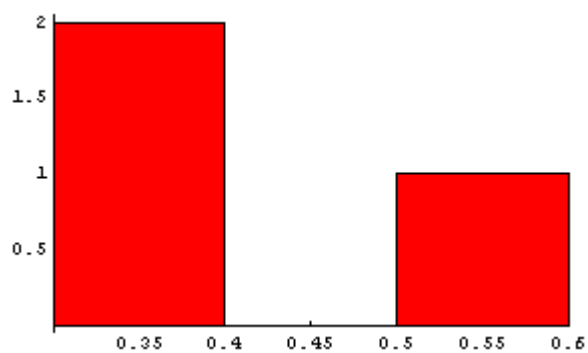
Obr. 30. Histogram pro systém (20)

Výsledky pro systém (22) jsou v příloze (P IV) a grafický průběh v příloze (P VIII). Histogram je na obrázku (Obr. 31.)



Obr. 31. Histogram pro systém (22)

Výsledky pro systém (23) jsou v příloze (P V) a grafický průběh v příloze (P IX). Histogram je na obrázku (Obr. 32.)



Obr. 32. Histogram pro systém (23)

ZÁVĚR

Diplomová práce představuje použití symbolické regrese ve spojení s evolučním algoritmem pro identifikaci neznámého systému bez znalostí jeho struktury.

Z grafů je vidět, že Analytické programování (symbolická regrese) je schopno vytvořit matematický model z neznámých dat. Z výsledků též vyplývá, že nalezené funkce je velmi dobře popisují. Výhodou analytického programování je jeho nezávislost na použitém evolučním algoritmu.

Analytické programování se dá použít také k řešení diferenciálních rovnic [5] a jiných problémů.

SEZNAM POUŽITÉ LITERATURY

- [1] Bobál Valdimír, Identifikace systémů, 1.vydání, 1990, ISBN 80-214-0125-7
- [2] Koza John R., Genetic Programming II; Automatic Discovery of Reusable Programs, MIT Press, 1998,
- [3] Koza John R., Genetic Programming, MIT Press, 1998, ISBN 0-262-11189-6
- [4] Kvasnička V., Pospíchal J., Tiňo P., Evolučné algoritmy, STU Bratislava, 2000, ISBN 80-227-1377-5
- [5] Zelinka I., Umělá inteligence v problémech globální optimalizace, BEN, Praha, 2002, ISBN 80-7300-069-5
- [6] Zelinka I., Programming by Means of Soma Algorithm. ICICIS'02, First International Conference on Intelligent Computing and Information Systems, Egypt, Cairo, 2002, ISBN 977-237-172-3
- [7] Zelinka I., Analytic Programming by Means of Soma Algorithm. Mendel '02, In: Proc. 8th International Conference on Soft Computing Mendel'02, Brno, Czech Republic, 2002, 93-101., ISBN 80-214-2135-5

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ε	Malé kladné číslo u GA
α	Chladicí koeficient, parametr SA
AllToAll	Verze Somy
AllToOne	Verze Somy
Cost function	Účelová funkce
Cost value	Hodnota účelové funkce
CR	Práh křížení, parametr DE
DE	Diferenciální evoluce
Dim	Dimenze systému, parametr DE a SOMA
F	Mutační konstanta, parametr DE
$F(\alpha)$	Fitness pro jedince α
f_{\max}	Maximální hodnota účelové funkce
f_{\min}	Minimální hodnota účelové funkce
GA	Genetická algoritmus
GFS	Obecný funkční prostor
K	Zesílení systému
N_{Eval}	Počet ohodnocení účelové funkce
PopSize	Velikost populace, parametr DE a SOMA
SA	Simulované žihání
SOMA	Samo-organizující se migrační algoritmus
Step	Krok mapování cesty jedince u SOMy
T	Časová konstanta systému

SEZNAM OBRÁZKŮ

Obr. 1. Schéma procesu identifikace	12
Obr. 2. Příklady testovacích signálů	13
Obr. 3. Přechodová charakteristika soustavy prvního řádu.....	15
Obr. 4. Lineární t-invariantní systém	17
Obr. 5. Odezva systému na čtvercovou funkci	17
Obr. 6. Impulsní charakteristika systému.....	18
Obr. 7. Porovnání nejlepšího jedince s originální impulsní funkcí.....	19
Obr. 8. Porovnání nejlepšího jedince s odezvou na čtvercovou funkci.	19
Obr. 9. Porovnání nejlepšího jedince s odezvou na rampu.....	20
Obr. 10. Porovnání nejlepšího jedince s odezvou na jednotkový skok.....	20
Obr. 11. Porovnání nejlepšího jedince s odezvou na krátkou čtvercovou funkci.....	20
Obr.12. Schéma ruletového kola na jednotkovém kruhu.....	22
Obr. 13. Schéma jednotkové úsečky	22
Obr. 14. Schéma křížení.....	23
Obr. 15. Schéma mutace	23
Obr. 16. Typický průběh Simulovaného žíhání	25
Obr. 17. Princip Diferenciální evoluce	26
Obr. 18. Princip SOMA	29
Obr. 19. Princip ošetření diskrétního parametru jedince	32
Obr.20.Obecný funkční prostor.....	33
Obr. 21. Rozdělení funkcí podle počtu argumentů	33
Obr. 22. Obecný funkční prostor GFS – příklad	34
Obr. 23. Mapování jedince – nepatologický případ.....	34
Obr. 24. Mapování jednice – patologické případ.....	34
Obr. 25. Účelová funkce v AP	35
Obr. 26. CriticalParameterSet.....	36
Obr. 27. CostValue funkce.....	39
Obr. 28. Definice funkce MainAP	40
Obr. 29. Histogram pro systém (19).....	42
Obr. 30. Histogram pro systém (20).....	42
Obr. 31. Histogram prosystém (22).....	42

Obr. 32. Histogram pro systém (23)..... 43

SEZNAM TABULEK

Tab. 1. Parametry DE.....	25
Tab. 2. Parametry SOMA	28
Tab. 3 Výsledky simulací statisticky.....	41

SEZNAM PŘÍLOH

P I Funkce AP v programu Mathematica

P II Výsledky pro $\frac{1}{s^2 + 2s + 1}$

P III Výsledky pro $\frac{1}{(s + 1)^3}$

P IV Výsledky pro $\frac{1 - 0,5s}{s(s + 1)}$

P V Výsledky pro $\frac{1}{(1+s)(1+0,5s)(1+0,5^2s)(1+0,5^3s)}$

P VI Grafický průběh pro $\frac{1}{s^2 + 2s + 1}$

P VII Grafický průběh pro $\frac{1}{(s + 1)^3}$

P VIII Grafický průběh pro $\frac{1 - 0,5s}{s(s + 1)}$

P IX Grafický průběh pro $\frac{1}{(1+s)(1+0,5s)(1+0,5^2s)(1+0,5^3s)}$

PŘÍLOHA P I: FUNKCE AP V PROGRAMU MATHEMATICA

MainAP funkce:

```

MainAP]:=Module[{},
  BasicArithmetic={
    {Plus,2,{{}}},{Subtract,2,{{}}},{Minus,1,{{}}},{Times,2,{{}}},{Divide,2,{{}},
    {Power,2,{{}}}};

  ElementaryFunctions={
    {Log,2,{{0,Infinity}}},{Exp,1,{{-Infinity,Infinity}}},
    {Power,2,{{-Infinity,Infinity}}},
    {Sqrt,1,{{-
    Infinity,Infinity}}},{Sin,1,{{}}},{Cos,1,{{}}},{Tan,1,{{}}},{Csc,1,{{}},
    {Sec,1,{{}}},{Cot,1,{{}}},{ArcSin,1,{{}},
    {ArcCos,1,{{}}},{ArcTan,1,{{}}},{ArcCsc,1,{{}},
    {ArcSec,1,{{}}},{ArcCot,1,{{}}},{Sinh,1,{{}}},{Cosh,1,{{}}},{Tanh,1,{{}},
    {Csch,1,{{}}},{Sech,1,{{}}},{Coth,1,{{}}},{ArcSinh,1,{{}}},{ArcCosh,1,{{}},
    {ArcTanh,1,{{}}},{ArcCsch,1,{{}}},{ArcSech,1,{{}}},{ArcCoth,1,{{}}}};

  VariousConstants={{Pi,0,{{}}},{t,0,{{}}},{E,0,{{}},
    {t,0,{{}}},{GoldenRatio,0,{{}}},{t,0,{{}}},{EulerGamma,0,{{}}},{t,0,{{}},
    {Catalan,0,{{}}},{t,0,{{}}},{Khinchin,0,{{}}},{t,0,{{}}},{Glaisher,0,{{}},
    {t,0,{{}}},{I,0,{{}}},{t,0,{{}}}};

  NumericalFunctions={{Abs,1,{{-Infinity,Infinity}}},{Sign,1,{{-
    Infinity,Infinity}}},
    {Round,1,{{Infinity,Infinity}}},{IntegerPart,1,{{-Infinity,Infinity}}},
    {FractionalPart,1,{{-Infinity,Infinity}}},{Floor,1,{{-Infinity,Infinity}}},
    {Ceiling,1,{{-Infinity,Infinity}}},{Chop,1,{{-Infinity,Infinity}}},
    {Max,1,{{-Infinity,Infinity}}},{Min,1,{{-Infinity,Infinity}}},
    {Re,1,{{-Infinity,Infinity}}},{Im,1,{{-Infinity,Infinity}}},
    {Conjugate,1,{{-Infinity,Infinity}}},{Arg,1,{{-Infinity,Infinity}}},
    {Mod,2,{{-Infinity,Infinity}}},{Quotient,2,{{-Infinity,Infinity}}}};

```

```

FactorialRelated={
  {Binomial,2,{{-Infinity},Infinity}}, {Gamma, 1, {{-Infinity},Infinity}},
  {Beta, 2, {{-Infinity},Infinity}}, {GammaRegularized, 2, {{-Infinity},Infinity}},
  {BetaRegularized, 3, {{-Infinity},Infinity}},
  {InverseGammaRegularized, 2, {{-Infinity},Infinity}},
  {InverseBetaRegularized, 3, {{-Infinity},Infinity}}, {LogGamma,1,{{0},Infinity}},
  {PolyGamma,1,{{-Infinity},Infinity}} };

AllFunctions= Flatten[ {BasicArithmetic,ElementaryFunctions,VariousConstants,
  NumericalFunctions,FactorialRelated},1];
ZeroArgumentFunctions=Position[AllFunctions,0,2];
OneArgumentFunctions=Position[AllFunctions,1,2];
TwoArgumentFunctions=Position[AllFunctions,2,2];
ThreeArgumentFunctions=Position[AllFunctions,3,2];

ZeroFunctions=AllFunctions[[#1[[1]]]&/@ZeroArgumentFunctions];
OneFunctions=AllFunctions[[#1[[1]]]&/@OneArgumentFunctions];
TwoFunctions=AllFunctions[[#1[[1]]]&/@TwoArgumentFunctions];
ThreeFunctions=AllFunctions[[#1[[1]]]&/@ThreeArgumentFunctions];
TwoOneZeroFunctions=Flatten[{ZeroFunctions,OneFunctions,TwoFunctions},1];
OneZeroFunctions=Flatten[{ZeroFunctions,OneFunctions},1];

AllDim=Dimensions[AllFunctions][[1]];
TwoOneZeroFunctions=GrowFunc[TwoOneZeroFunctions];
OneZeroFunctions=GrowFunc[OneZeroFunctions];
ZeroFunctions=GrowFunc[ZeroFunctions];
]

```

PickUp funkce:

```
PickUpFunction[pos_/,pos>=3]:=Module[{},
  posit++;
  freeposit=AllFunctions[[Individual[[posit]],2]];
  Return[{AllFunctions[[Individual[[posit]],1]],
    AllFunctions[[Individual[[posit]],2]]}]
]
PickUpFunction[pos_/,pos\[Equal]2]:=Module[{},
  posit++;
  freeposit=TwoOneZeroFunctions[[Individual[[posit]],2]];
  Return[{TwoOneZeroFunctions[[Individual[[posit]],1]],
    TwoOneZeroFunctions[[Individual[[posit]],2]]}]
]
PickUpFunction[pos_/,pos\[Equal]1]:=Module[{},
  posit++;
  freeposit=OneZeroFunctions[[Individual[[posit]],2]];
  Return[{OneZeroFunctions[[Individual[[posit]],1]],
    OneZeroFunctions[[Individual[[posit]],2]]}]
]
PickUpFunction[pos_/,pos\[Equal]0]:=Module[{},
  posit++;
  Return[{ZeroFunctions[[Individual[[posit]],1]],
    ZeroFunctions[[Individual[[posit]],2]]}]
]
```

Give funkce:

```

GiveFunction[ind_]:=Module[{},
  Individual=ind;
  dm=Dimensions[ind][[1]];
  freeposit=dm-1;
  posit=0;
  tbl=Table[PickUpFunction[freeposit],{dm}];

  pp=Position[tbl,K];
  pocParm=Dimensions[pp][[1]];
  Table[tbl[[pp[[i,1]],pp[[i,2]]]]=K[[i]],{i,pocParm}];

  posit=1;
  tbl2=Flatten[{
    If[#1[[2]]\{NotEqual}0,
      {#1[[1]],Table[i,{i,posit+1,posit+=#1[[2]]}],{#1[[1]],{0}}
    ]
  },1]&/@tbl;
  tbl3=Table[
    If[tbl2[[i,2,1]]>0,tbl2[[i]]={
temp=tbl2[[#1]][[1]]&/@tbl2[[i,2]];tbl2[[i,1]]@@temp/.Infinity\[Rule]1000000.,
    0},tbl2[[i]]
  ],{i,dm,1,-1}];
  Return[{tbl3[[dm,1]],Dimensions[Position[tbl3[[dm,1],K]][[1]]]}
]

```

PŘÍLOHA P II: VÝSLEDKY PRO $\frac{1}{s^2 + 2s + 1}$

373

0.0366449

$$\left\{ \frac{K[1]^2}{(-s + 3K[2]) \left(-K[3] - \frac{s}{2K[4]} \right)}, 4 \right\}$$

$$10.8569 (-0.318002 (3 - 3e^{-1.04617t}) - 1.04617 (-1 + e^{-0.954005t}))$$

113

2.37731 × 10⁻¹³

$$\left\{ -\frac{2K[1]}{(s + K[2])^2}, 2 \right\}$$

$$1. (1 - e^{-1 \cdot t} (1 + 1 \cdot t))$$

126

0.144874

$$\left\{ -\frac{(2s - K[2]) (-s - sK[3])}{s (s + K[1])^2}, 3 \right\}$$

$$-0.026956 e^{-1.01507t} t + 0.998723 (1 - e^{-1.01507t} (1 + 1.01507t))$$

395

0.000166814

$$\left\{ \frac{K[3] + K[4]^2 + \frac{K[5]}{s} - s \left(1 + \frac{-s+K[6]}{s} \right) + s^2 K[7]}{(-s + K[1]) (s + K[2])}, 7 \right\}$$

$$56.3712 (0.0177399 + 0.991177 e^{-1.00892t} - 1.00892 e^{-0.991177t}) + \\ 0.000195041 (-e^{-1.00892t} + e^{-0.991177t}) - \\ 0.000138772 (1.01788 (-1 + e^{-0.991177t}) + 1.0089t - 0.982401 (-1 + e^{-1.00892t} + 1.00892t)) - \\ 1.72993 \times 10^{-6} \text{UnitStep}[t]$$

1966

0.0452917

$$\left\{ \frac{K[1] (s - K[3] - K[4]^2)}{(-s - K[2]) \left(-s + 2K[5] + \frac{2s}{K[6]} \right)}, 6 \right\}$$

$$-0.0479751 e^{-1.17495t} (-1 + e^{0.297977t}) - \\ 1.97608 e^{-1.17495t} (1.48994 (-1 + e^{1.17495t}) - 1.99619 (-e^{0.297977t} + e^{1.17495t}))$$

45

0.47335688207080084

$$\left\{ K[1] - \frac{-s + K[3]^2}{(-s - K[2]) (3s - 2K[4])}, 4 \right\}$$

$$1.00517 + 0.715665 e^{-2.10802t} - 1.6779 e^{-0.700459t}$$

130

 1.57037×10^{-10}

$$\left\{ -\frac{K[1] (K[2] - K[3] (-s + K[4])) \left(\frac{s}{K[7]} - s K[8] \right)}{s (-2s - K[5]) (-s - K[6])}, 8 \right\}$$

 $-1.50744 (-0.663377 - 38179.5 e^{-1.00001t} + 38180.1 e^{-0.999991t})$

31

0.0000197956

$$\left\{ \frac{\frac{1}{2} (-s - K[2]) + \frac{K[3]}{s^2} - 2s^2 K[4]}{s K[1] (-s - s^2 + \frac{K[5] (s+K[6])}{s})}, 6 \right\}$$

0.999983 (1.00002 -

 $38901.2 (5.00016 e^{-1.00001t} + 1.00002 e^{-1 \cdot t} - 2.00005 e^{1.00001t} + 1.00002 (e^{-1.00001t} - e^{-1 \cdot t}) -$
 $1.00002 (e^{-1.00001t} + e^{-1 \cdot t}) - 2.00005 (e^{-1.00001t} + e^{-1 \cdot t} - 2 e^{1.00001t}) -$
 $1.00004 (e^{-1.00001t} - e^{1.00001t}) - 1.00003 (2 e^{-1.00001t} - e^{-1 \cdot t} - e^{1.00001t}) -$
 $1.00003 (e^{-1.00001t} + e^{1.00001t}) - 3.00009 (e^{-1.00001t} - 2 e^{-1 \cdot t} + e^{1.00001t})) -$
 $4.85354 \times 10^{-8} t + 2.42677 \times 10^{-8} t^2)$

722

0.

$$\left\{ \frac{-K[1] + s (-1 - K[2])}{1 + 2s + s^2}, 2 \right\}$$

 $1 \cdot e^{-t} (-1 + e^t - t) + 0 \cdot e^{-t} t$

53

 2.30742×10^{-10}

$$\left\{ -\frac{1 + s + s K[3] - \frac{2s - K[6]}{K[4] (s+K[5])}}{K[1] (2s^2 + s K[2]^3)}, 6 \right\}$$

 $1.67201 \times 10^{-6} (e^{-1.00001t} - e^{-0.999986t}) -$
 $9040.17 (1.99997 (2 - 2 e^{-1.00001t}) + 4.00006 (-1 + e^{-0.999986t})) -$
 $1.9787 \times 10^{-8} (8.00022 (-1 + e^{-0.999986t}) + 3.99989 (2 - 2 e^{-1.00001t} - 2.00003t) + 8.00011t)$

1161

 3.6317×10^{-10}

$$\left\{ -\frac{2s K[1] (1 - K[2] + K[3]^2)}{(s + K[4]) (-s + 2s (s + K[5]))}, 5 \right\}$$

 $59560.8 (0.999992 (-1 + e^{-1.00001t}) + 0.500004 (2 - 2 e^{-0.999992t}))$

517

 1.24569×10^{-11}

$$\left\{ -\frac{-1 - K[1] - \frac{s - K[3]}{-s + K[2]}}{1 + 2s + s^2}, 3 \right\}$$

 $0.249978 (e^{1.00009t} - e^{-t} (1 + 2.00009t)) -$
 $0.249978 e^{-t} (e^t (-1 + e^{1.00009t}) + 1.00009 (2 - 2e^t + t + 1.00009 (1 - e^t + t)))$

3358

 7.20528×10^{-8}

$$\left\{ -\frac{K[1] \left(s + K[2] - (-K[3] + s K[4]) \left(K[5] + \frac{K[6]}{1+s+K[7]} \right) \right)}{1+s}, 7 \right\}$$

$$-6.72554 \times 10^7 e^{-t} \left(1 - 1. e^{-1.48687 \times 10^{-8} t} - 1.48687 \times 10^{-8} e^t \right) + 1.06746 e^{-2 \cdot t} (e^t - e^{1 \cdot t}) + 3.17371 \times 10^{-9} \text{UnitStep}[t]$$

89

 3.78959×10^{-10}

$$\left\{ -\frac{s + K[3]}{K[1] (s - K[2]) \left(-1 - \frac{s}{K[4]} \right) (2s + 2s K[5])}, 5 \right\}$$

$$34009. (0.000029404 + 0.999985 e^{-1.00001 t} - 1.00001 e^{-0.999985 t}) - 2.3884 \times 10^{-7} (1.00003 (-1 + e^{-0.999985 t}) + 1.00001 t - 0.999971 (-1 + e^{-1.00001 t} + 1.00001 t))$$

359

0.0314554

$$\left\{ -\frac{K[1]^2 (s - K[2])}{-1 - \frac{s}{K[3]}} - \frac{s + K[4]}{2 K[5] + \frac{s}{K[6]}}, 6 \right\}$$

$$-0.039072 (-e^{-1.14892 t} + e^{-0.888052 t}) - 3.83069 (0.888052 (1 - e^{-1.14892 t}) + 1.14892 (-1 + e^{-0.888052 t})) + 0.00113301 \text{UnitStep}[t]$$

2472

0.892707

$$\left\{ K[1] + \frac{K[2]}{2s^4} - \frac{-K[3] + \frac{K[4]}{2s}}{2s - K[5]}, 5 \right\}$$

$$1.62028 (-1 + e^{0.260734 t}) + 1.37993 (2 - 2 e^{0.260734 t} + 0.521468 t) + 0.0887241 (48 (-1 + e^{0.260734 t}) - 0.521468 t (24 + 0.521468 (6 + 0.521468 t) t)) + 0.0110905 (-384 (-1 + e^{0.260734 t}) + 0.521468 t (192 + 0.521468 t (48 + 0.521468 (8 + 0.521468 t) t))) - 0.0863212 \text{UnitStep}[t]$$

700

0.0189077

$$\left\{ \frac{K[1]^3 (-s - 2 K[2]) \left(s + \frac{K[4]^3}{s+K[5]} \right)}{s (s - K[3])}, 5 \right\}$$

$$2.27675 (e^{-0.999184 t} - e^{-0.998469 t}) - 1396.15 (-0.000715379 - 0.998469 e^{-0.999184 t} + 0.999184 e^{-0.998469 t}) + 0.105364 (1.00163 (-1 + e^{-0.999184 t}) - 1.00307 (-1 + e^{-0.998469 t} + 0.998469 t) + 1.00082 t) + 0.00162994 \text{UnitStep}[t]$$

59

0.000469392

$$\left\{ -\frac{K[1] (s - K[4] - s K[5])}{s (-s - 2 K[2]) (-s + K[3])}, 5 \right\}$$

$$15.4478 (1.96783 (-1 + e^{-1.01635 t}) - 2.03257 (-1 + e^{-0.983975 t})) + 0.000259177 (0.968207 (-1 + e^{-1.01635 t}) + 0.25824 (4 - 4 e^{-0.983975 t} - 3.9359 t) + 0.984033 t)$$

950

$$1.43678 \times 10^{-14}$$

$$\left\{ \frac{K[1]^2 (s - s (s + K[4]))}{s (-s + K[2])^2 \left(\frac{1}{2} - \frac{s}{K[3]} \right)}, 4 \right\}$$

$$0.186355 (4 e^{0.158241 t} - 2 e^{-1 \cdot t} (2 + 2.31648 t)) - \\ 0.0931777 (0.200322 (-1 + e^{-1 \cdot t}) - 0.632965 (4 - e^{-1 \cdot t} (4 + 0.316483 t))) + \\ 4 \cdot (-2 + 2 e^{0.158241 t} + 0.316483 e^{-1 \cdot t} t)$$

138

$$0.00609938$$

$$\left\{ -\frac{(s + 2 K[3]) (2 s + K[4])}{4 s K[1]^2 (-s + K[2])^2}, 4 \right\}$$

$$0.999665 (1 + e^{-1.00108 t} (-1 - 1.00108 t)) - \\ 0.0000487781 (2 + e^{-1.00108 t} (-2 - 1.00108 t) - 1.00108 t) - 0.00133482 e^{-1.00108 t} t$$

1328

$$0.40155$$

$$\left\{ -\frac{K[1] (1 - s - K[4]) (2 s - 2 K[6])}{(-1 - s) s (K[2] + K[3]^2) (s - K[5])}, 6 \right\}$$

$$0.452866 e^{-t} (-1 + e^{0.126544 t}) + 7.0799 (-1 + e^{-t} - 1.14488 (-1 + e^{-0.873456 t})) - \\ 0.0344892 (-1.14488 (1 - e^{-0.873456 t}) + t - 0.873456 (-1 + e^{-t} + t))$$

Pozn.: Výsledky v příloze jsou ve formátu:

Počet ohodnocení účelové funkce

Hodnota účelové funkce

Nefitovaný tvar funkce

Fitovaný tvar funkce

PŘÍLOHA P III: VÝSLEDKY PRO $\frac{1}{(s+1)^3}$

1407

0.755882

$$\left\{ -K[[1]] + \frac{s(K[[2]] - \frac{K[[3]]}{s})}{s + K[[4]]} - K[[5]] \left(K[[6]] + \frac{K[[7]]^3}{s + 2K[[8]]} \right), 8 \right\}$$

$$6.76147 (e^{-0.759862t} - e^{-0.719311t}) + 24.0781 (0.0405516 + 0.719311 e^{-0.759862t} - 0.759862 e^{-0.719311t}) + 0.0370063 \text{UnitStep}[t]$$

2151

0.923486

$$\left\{ \frac{K[[1]] (1 + s + K[[3]]) \left(s + s^2 - \frac{s(-s + K[[5]] - K[[6]]^2)}{K[[4]]} \right)}{(s - K[[2]]) \left(\frac{s^2}{K[[7]]} + 2s(s + K[[8]]) \right)}, 8 \right\}$$

$$2241.21 e^{-0.720807t} (-1 + e^{-0.0000834435t}) - 6331.11 (-1.38717 (-1 + e^{-0.720891t}) - 1.38733 e^{-0.720807t} (-1 + e^{0.720807t})) - 0.000550828 \text{UnitStep}[t]$$

142

0.753568

$$\left\{ \frac{K[[1]] \left(-s + K[[4]] + \frac{s}{K[[5]]} + 2sK[[6]] - \frac{sK[[7]]}{s + K[[8]]} \right)}{s + K[[2]] + sK[[3]]}, 8 \right\}$$

$$-1.53773 \times 10^9 e^{-1.47827t} (-e^{0.739134t} + e^{-0.739134t}) + 3.53876 \times 10^9 (1.14323 (-1 + e^{-0.739134t}) + 1.14323 (1 - e^{-0.739134t})) + 0.0372844 \text{UnitStep}[t]$$

129

0.918942

$$\left\{ \frac{K[[1]] (2 - s + K[[4]] + K[[5]]^2)}{2(-s - K[[2]])(-s + K[[3]])}, 5 \right\}$$

$$48545.7 (e^{-0.721226t} - e^{-0.721222t}) + 261701. (3.88253 \times 10^{-6} + 0.721222 e^{-0.721226t} - 0.721226 e^{-0.721222t})$$

1061

0.918944

$$\left\{ \frac{-\frac{1}{K[[4]]} + \frac{s + sK[[6]]^2}{s + K[[5]]}}{-2K[[1]] - \frac{s + K[[3]]}{K[[2]]}}, 6 \right\}$$

$$426736. (e^{-0.721224t} - e^{-0.721223t}) + 1.19662 \times 10^6 (1.38653 (-1 + e^{-0.721224t}) + 1.38653 (1 - e^{-0.721223t}))$$

175

0.494889

$$\left\{ K[1] \left(K[2] - \frac{-K[3]^2 - \frac{s(-s+K[5] - \frac{2K[6]}{s})}{K[4](s-K[7])(3s-K[8])}}{s} \right), 8 \right\}$$

$$\begin{aligned} & -72478.1 (-e^{-0.671309t} + e^{-0.671306t}) + \\ & 42289.9 (6.04175 (-1 + e^{-0.671309t}) - 6.04178 (-1 + e^{-0.671306t})) + \\ & 1274.94 (-4.0559 (-1 + e^{-0.671306t}) - 2.72275 t + \\ & 0.450652 (-9 + 9e^{-0.671309t} + 6.04178 t) + 0.0253971 \text{UnitStep}[t] \end{aligned}$$

370

0.822623

$$\left\{ -\frac{K[1] \left(-s - \frac{K[4] - \frac{K[5]}{s} + sK[6]}{(s+K[2]) \left(s + \frac{K[3]^2}{s} \right)} \right)}{s}, 6 \right\}$$

$$\begin{aligned} & 6.08177 \times 10^8 (1.36842 \times 10^{-10} (-1 + e^{-0.663156t}) + \\ & 9.07474 \times 10^{-11} t + 0.000226623 (-1 + \text{Cos}[0.000515315 t]) + \\ & 0.29164 (0.000515315 t - \text{Sin}[0.000515315 t])) - \\ & 1.37684 \times 10^6 (2.6555 \times 10^{-7} (1 - e^{-0.663156t}) - \\ & 0.439776 (-1 + \text{Cos}[0.000515315 t]) - 0.000341734 \text{Sin}[0.000515315 t]) + \\ & 1494.13 (0.000515315 e^{-0.663156t} - 0.000515315 \text{Cos}[0.000515315 t] + \\ & 0.663156 \text{Sin}[0.000515315 t]) - 0.0270914 \text{UnitStep}[t] \end{aligned}$$

1251

0.772729

$$\left\{ -1 + 2K[1] + \frac{K[2]}{s+K[3]} + \frac{s+K[4]}{s+2K[5]}, 5 \right\}$$

$$\begin{aligned} & -7.48432 (-0.130446 - 0.67853 e^{-0.808976t} + 0.808976 e^{-0.67853t}) - \\ & 2.12144 (-e^{-0.808976t} + e^{-0.67853t}) + 0.0375051 \text{UnitStep}[t] \end{aligned}$$

56

0.525207

$$\left\{ K[1] + \frac{K[2](-s+K[3])}{s(2s-K[4]^3)} + \frac{2K[6]+2s^2K[7]}{s(-2s-2K[5])}, 7 \right\}$$

$$\begin{aligned} & -0.4046 (-2e^{-0.81156t} + 2e^{-0.553091t}) + \\ & 0.491961 (-2.38364 (2 - 2e^{-0.81156t}) - 6.99512 (-1 + e^{-0.553091t})) - \\ & 0.0182352 (5.26904 (-1 + e^{-0.553091t}) + \\ & 1.22364 (2 - 2e^{-0.81156t} - 1.62312 t) + 2.91426 t) + 0.0256385 \text{UnitStep}[t] \end{aligned}$$

279

0.753571

$$\left\{ -\frac{K[1](K[4]+2sK[5]^2(-s-K[6]))}{(s+2K[2])(-1+s+4K[3])}, 6 \right\}$$

$$\begin{aligned} & 173077. (e^{-0.739135t} - e^{-0.739133t}) - \\ & 168281. (2.70587 (1 - e^{-0.739135t}) + 2.70587 (-1 + e^{-0.739133t})) + \\ & 0.0372848 \text{UnitStep}[t] \end{aligned}$$

824

0.935361

$$\left\{ -K[1] - \frac{K[2] \left(2s + K[3] + \frac{K[4]}{s} \right) (s - K[5] + sK[6])}{3s^2 (s + s^2)}, 6 \right\}$$

0.38061 $(-1 + e^{-t} + t) -$

$$0.0707651 \left(1 - e^{-t} - t + \frac{t^2}{2} \right) - 0.00618648 \left(-1 + e^{-t} + t - \frac{t^2}{2} + \frac{t^3}{6} \right) +$$

$$0.00446953 \left(\frac{1}{24} (24 + t(-24 + t(12 + (-4 + t)t))) - \text{Cosh}[t] + \text{Sinh}[t] \right) -$$

0.0331809 UnitStep[t]

1385

0.753572

$$\left\{ K[1] - \frac{2(2K[5] + sK[6])(-s + sK[7])}{s(s + 2K[2])(K[3] + sK[4])}, 7 \right\}$$

91147.9 $(e^{-0.739135t} - e^{-0.739132t}) -$

$$88622.2(2.70586(1 - e^{-0.739135t}) + 2.70588(-1 + e^{-0.739132t})) +$$

0.0372845 UnitStep[t]

1896

0.691897

$$\left\{ \frac{K[1]^2 \left(1 + s + K[3] + \frac{K[4]}{-s+K[5]} - s(s + K[6]) \right)}{2s(s + K[2])}, 6 \right\}$$

1.19277 $(e^{-0.683195t} - e^{-0.587226t}) +$

$$12.0139(0.095969 + 0.587226e^{-0.683195t} - 0.683195e^{-0.587226t}) +$$

$$0.526468(0.344835(-1 + e^{-0.683195t}) +$$

$$0.466756(1 - e^{-0.587226t} - 0.587226t) + 0.23559t) - 0.0106349 \text{UnitStep}[t]$$

2188

0.711056

$$\left\{ -\frac{K[1] \left(s - K[3] + \frac{K[4]}{s} \right) \left(-1 - s + K[5] - \frac{s}{K[6]} \right)}{s^3 (-2s + K[2])}, 6 \right\}$$

-1.44479 $(2 - 2e^{0.302204t} + 0.604409t) +$

$$1.15307(8 - 8e^{0.302204t} + 2.41763t + 0.36531t^2) +$$

$$0.258431(48(-1 + e^{0.302204t}) - 0.604409t(24 + 0.604409(6 + 0.604409t)t)) +$$

$$0.0186075(-384(-1 + e^{0.302204t}) +$$

$$0.604409t(192 + 0.604409t(48 + 0.604409(8 + 0.604409t)t)))$$

2432

0.760932

$$\left\{ \frac{s}{K[1](2K[2] + sK[3])} - K[4] \left(-K[5] - \frac{K[6]}{1 + sK[7]} \right), 7 \right\}$$

3.13353 $(e^{-0.785343t} - e^{-0.697279t}) -$

$$3.03311(2.54666(1 - e^{-0.785343t}) + 2.86829(-1 + e^{-0.697279t})) +$$

0.0380323 UnitStep[t]

```
2027
0.799925
{ - 2 K[[1]] (-s + K[[2]]) (K[[4]] + s K[[5]]) / (s + K[[3]])^2 , 5 }
0.993454 (1 - e-0.73408 t (1 + 0.73408 t)) -
0.241978 e-0.73408 t t + 0.0205312 UnitStep[t]
```

Pozn.: Výsledky v příloze jsou ve formátu:

Počet ohodnocení účelové funkce

Hodnota účelové funkce

Nefitovaný tvar funkce

Fitovaný tvar funkce

PŘÍLOHA P IV: VÝSLEDKY PRO $\frac{1-0,5s}{s(s+1)}$

164

 5.44564×10^{-14}

$$\left\{ -\frac{(s - K[2]) (-s - s K[3])}{(1 + s) K[1] (s^2 + K[4])}, 4 \right\}$$

$$-0.5 (-e^{-t} + \text{Cos}[4.96542 \times 10^{-9} t] + 4.96542 \times 10^{-9} \text{Sin}[4.96542 \times 10^{-9} t]) + 1. (e^{-t} - \text{Cos}[4.96542 \times 10^{-9} t] + 2.01393 \times 10^8 \text{Sin}[4.96542 \times 10^{-9} t])$$

461

0.364061

$$\left\{ -\frac{K[1] \left(-\frac{1}{s} - 2 K[2] \right) (-s + K[3]) \left(-s - 2 K[4] - \frac{K[5]}{s} \right)}{s (s + K[6])}, 6 \right\}$$

$$-0.54052 (1 - e^{-1.18554t}) + 0.772591 (-1 + e^{-1.18554t} + 1.18554 t) + 0.0101839 (2 - 2 e^{-1.18554t} + 1.18554 t (-2 + 1.18554 t)) - 0.000462926 (-6 + 6 e^{-1.18554t} + 1.18554 t (6 + 1.18554 t (-3 + 1.18554 t))) + 0.0394171 \text{UnitStep}[t]$$

32

0.543382

$$\left\{ \frac{K[1] \left(1 - s - 3 K[2] + \frac{K[3]}{-s+K[4]} + \frac{s}{K[5]} \right)}{1 + s}, 5 \right\}$$

$$-0.580981 e^{-t} (-1 + e^{0.998212t}) + 1.01111 (-1 + e^{-t} - 559.156 (-1 + e^{-0.00178841t})) + 0.053027 \text{UnitStep}[t]$$

305

 7.19046×10^{-14}

$$\left\{ K[1] + \frac{K[2] \left(2s - 2K[3] + \frac{-2+s}{K[4]} + \frac{K[5]}{2s} \right)}{s + s^2}, 5 \right\}$$

$$-0.5 (1 - e^{-t}) + 1. (-1 + e^{-t} + t) - 3.80576 \times 10^{-17} \left(1 - e^{-t} - t + \frac{t^2}{2} \right) + 1.42576 \times 10^{-16} \text{UnitStep}[t]$$

196

 7.29278×10^{-14}

$$\left\{ \frac{-s - K[2]}{s K[1]^2 (s - K[3])}, 3 \right\}$$

$$0.5 (-1 + e^{-1 \cdot t}) + 1. (-1 + e^{-1 \cdot t} + 1 \cdot t)$$

262

 1.67643×10^{-13}

$$\left\{ \frac{K[1] \left(-1 - 2 K[2] + s K[3]^3 + \frac{s}{-K[4]+K[5] (s+K[6] (s-K[7]))} \right)}{s}, 7 \right\}$$

$$0.5 (-1 + e^{-1 \cdot t}) + 1.14381 (0.874274 (-1 + e^{-1 \cdot t}) + 0.874274 t) - 1.08431 \times 10^{-16} \text{UnitStep}[t]$$

113

 5.2894×10^{-12}

$$\left\{ -K[1] - \frac{-4s + 3K[2]}{s} + \frac{-s - K[4] + sK[5] + \frac{s}{K[6](s-K[7])}}{s + K[3]}, 7 \right\}$$

$$-8.89771 (1.10696 - 0.106959 e^{-1 \cdot t} - 1 \cdot e^{0.106959 t}) +$$

$$0.451688 (e^{-1 \cdot t} - e^{0.106959 t}) + 8.44602 (0.0114402 (-1 + e^{-1 \cdot t}) +$$

$$1 \cdot (1 - e^{0.106959 t} + 0.106959 t) + 0.0114402 t) + 0. \text{UnitStep}[t]$$

443

0.0738241

$$\left\{ -\frac{K[1] \left(-2K[2] + \frac{K[3]}{s} \right) (-s - K[4])}{4s(-s + K[5])(s + K[6])}, 6 \right\}$$

$$-0.011241 (43.3118 + 1.01968 e^{-44.3314 t} - 44.3314 e^{-1.01968 t}) +$$

$$1.04045 (0.961767 (-1 + e^{-1.01968 t}) + 0.980697 t -$$

$$0.000508834 (-1 + e^{-44.3314 t} + 44.3314 t)) +$$

$$2.65112 \times 10^{-9} (-174247 \cdot (-1 + e^{-1.01968 t}) - 177677 \cdot t + 90586.9 t^2 -$$

$$1.06022 (2 - 2 e^{-44.3314 t} + 44.3314 t (-2 + 44.3314 t)))$$

801

0.685508

$$\left\{ -\frac{(s - 4K[3])(-s + sK[5])}{2s^2K[1](-s + K[2])(2s - K[4])}, 5 \right\}$$

$$-0.0246822 (4.51156 (-1 + e^{-5.15345 t}) - 20.6138 (-1 + e^{-1.12789 t})) -$$

$$0.0106531 (-106.232 (-1 + e^{-1.12789 t}) -$$

$$119.818 t + 1.27214 (-4 + 4 e^{-5.15345 t} + 20.6138 t))$$

141

 3.99032×10^{-11}

$$\left\{ K[1] + \frac{s + K[3]}{s - \frac{1}{K[2]}} + \frac{K[4] - K[5] \left(K[6] + \frac{K[7]}{-s - K[8]} \right) + \frac{K[9](s - K[10])}{s}}{s}, 10 \right\}$$

$$-0.251345 e^{-1 \cdot t} (-1 + e^{1.9893 t}) - 0.751345 (1 - e^{-1 \cdot t} - 1.01082 (-1 + e^{0.9893 t})) -$$

$$0.497311 e^{-1 \cdot t} (-1 + e^{1 \cdot t} (1 - 1 \cdot (t + 1.01082 (-1.01082 (-1 + e^{0.9893 t}) + t)))) +$$

$$5.53996 \times 10^{-14} e^{-1 \cdot t}$$

$$(2 - e^{1 \cdot t} (2 + 1 \cdot (t^2 + 1.01082 (t^2 + 2.02163 (-1.01082$$

$$(-1 + e^{0.9893 t} + t)))))) - 3.20277 \times 10^{-13} \text{UnitStep}[t]$$

252

 2.93858×10^{-10}

$$\left\{ \frac{-s^2 + sK[2] + s^2K[3]}{s^2(-1 - s - 2K[1])}, 3 \right\}$$

$$-0.5 (1 - e^{-1 \cdot t}) + 1 \cdot (-1 + e^{-1 \cdot t} + 1 \cdot t)$$

2

 5.48734×10^{-13}

$$\left\{ K[1] + \frac{K[4] + \frac{K[5]}{s(s - K[6])} - \frac{s - K[7]}{2s}}{-s - K[2] - 2sK[3]}, 7 \right\}$$

$$-0.423032 e^{-1 \cdot t} (-1 + e^{1.18194 t}) +$$

$$0.585693 e^{-1 \cdot t} (-1.57597 (-1 + e^{1 \cdot t}) + 8.66184 (-e^{1 \cdot t} + e^{1.18194 t})) - 0.0619792$$

$$e^{-1 \cdot t} (2.48367 (-1 + e^{1 \cdot t}) + 75.0275 e^{1 \cdot t} (-1 + e^{0.181944 t} - 0.181944 t) -$$

$$2.48367 e^{1 \cdot t} t) - 8.28469 \times 10^{-16} \text{UnitStep}[t]$$

15

 7.77902×10^{-10}

$$\left\{ K[1] + \frac{s + K[3] - s(s + K[4])}{-2s - s^2 K[2]}, 4 \right\}$$

$$-0.5(1 - e^{-1 \cdot t}) + 0.5(2 \cdot (-1 + e^{-1 \cdot t}) + 2t) + 4.61505 \times 10^{-11} \text{UnitStep}[t]$$

20

 1.94109×10^{-12}

$$\left\{ K[3] - \frac{K[4](2s + K[6])}{s - K[5]}, 6 \right\}$$

$$0.5(-1 + e^{-1 \cdot t}) + 1 \cdot (-1 + e^{-1 \cdot t} + 1 \cdot t) - 5.03318 \times 10^{-17} \text{UnitStep}[t]$$

45

0.40667

$$\left\{ \frac{K[1](1 - 2K[4] - s(-s - K[5]))}{s(-K[2] - \frac{1}{2}sK[3])}, 5 \right\}$$

$$-0.410395(1 - e^{-0.948197t}) + 0.210496(5.01921(-1 + e^{-0.948197t}) + 4.7592t) - 0.0479465 \text{UnitStep}[t]$$

9

0.427781

$$\left\{ -\frac{-1 + s - K[2]}{s(1 - s + K[1])} + \frac{-\frac{K[4]^2}{s} - K[5](s + 2K[6])}{s + K[3]}, 6 \right\}$$

$$-0.13437(-e^{-0.961558t} + e^{2 \cdot t}) + 0.59342(1.03998(-1 + e^{-0.961558t}) + 0.5(-1 + e^{2 \cdot t})) - 0.175579e^{-0.961558t}(-4.00001 + e^{0.961558t} - e^{0.961558t}(1 \cdot (-1 + 0.961558t) + 1 \cdot (-2 + 2.84771t) + 0.961558(t + 0.961558(1 - e^{2 \cdot t} + t)))) - 0.0488066 \text{UnitStep}[t]$$

46

 3.07254×10^{-14}

$$\left\{ \frac{-\frac{K[2] + K[3]^2}{K[1]} + \frac{K[4]}{s}}{-s + K[5]}, 5 \right\}$$

$$0.5(-1 + e^{-1 \cdot t}) + 1 \cdot (-1 + e^{-1 \cdot t} + 1 \cdot t)$$

79

 8.05606×10^{-14}

$$\left\{ \frac{2(2s - (-2K[2] - \frac{K[3]}{s})(-s + K[4]))}{s(s + K[1])}, 4 \right\}$$

$$-0.5(1 - e^{-1 \cdot t}) + 1 \cdot (-1 + e^{-1 \cdot t} + 1 \cdot t) - 1.81046 \times 10^{-17}(2 - 2e^{-1 \cdot t} + 1 \cdot (-2 + 1 \cdot t)t)$$

42

 8.18857×10^{-14}

$$\left\{ \frac{\frac{K[2]}{s} - \frac{K[4] - 2sK[5]^2 - K[6](-\frac{K[7]}{s^2} - \frac{s + 3K[8]}{s})}{K[3]}}{-s - K[1]}, 8 \right\}$$

$$-0.5(1 - e^{-1 \cdot t}) + 1 \cdot (-1 + e^{-1 \cdot t} + 1 \cdot t) - 5.58066 \times 10^{-17}(2 - 2e^{-1 \cdot t} + 1 \cdot (-2 + 1 \cdot t)t) - 2.88289 \times 10^{-15} \text{UnitStep}[t]$$

271

$$1.30467 \times 10^{-12}$$

$$\left\{ -\frac{s + K[1]}{2s(K[2] + sK[3])}, 3 \right\}$$

$$-0.5(1 - e^{-1 \cdot t}) + 1 \cdot (1 \cdot (-1 + e^{-1 \cdot t}) + 1 \cdot t)$$

26

$$8.11546 \times 10^{-14}$$

$$\left\{ K[1] + \frac{-2s - 2K[4] + sK[5] + \frac{1+s-K[6]}{2s}}{-1 - K[2] - sK[3]}, 6 \right\}$$

$$-0.5(1 - e^{-1 \cdot t}) +$$

$$0.381762(2.61944(-1 + e^{-1 \cdot t}) + 2.61944t) + 3.39072 \times 10^{-16} \text{UnitStep}[t]$$

256

$$3.76019 \times 10^{-14}$$

$$\left\{ -\frac{s^2 - sK[3] - s^2K[4]^2}{s^2K[1](s + K[2])}, 4 \right\}$$

$$-0.5(1 - e^{-1 \cdot t}) + 1 \cdot (-1 + e^{-1 \cdot t} + 1 \cdot t)$$

Pozn.: Výsledky v příloze jsou ve formátu:

Počet ohodnocení účelové funkce

Hodnota účelové funkce

Nefitovaný tvar funkce

Fitovaný tvar funkce

PŘÍLOHA P V: VÝSLEDKY PRO $\frac{1}{(1+s)(1+0,5s)(1+0,5^2s)(1+0,5^3s)}$

39

0.512158

$$\left\{ -\frac{K[[4]] * (s - K[[5]]) - \frac{K[[6]]}{3*s+K[[7]]}}{K[[1]] * \left(-s + \frac{s}{K[[2]]} + 2 * (2*s + K[[3]])\right)}, 7 \right\}$$

$$(-7.9684844680557765 * (-E^{(1.1658702934806908 * t)} + E^{(1.1965943347498151 * t)}) / E^{(2.362464628230506 * t)} - 23.14464304070799 * (0.2785683699589192 * (6 - 6 / E^{(1.1965943347498151 * t)}) + 1.7154566946113925 * (-1 + E^{(-1.1658702934806908 * t)})) - 0.017088665087706425 * UnitStep[t])$$

38

0.374094

$$\left\{ \frac{K[[1]]^3 (K[[6]] + K[[7]] (s + K[[8]]))}{(K[[2]] + s K[[3]]) (3s + K[[4]] + s K[[5]])}, 8 \right\}$$

$$-22248.7 (-e^{-1.19907t} + e^{-1.19905t}) - 77441.4 (1.0781 (1 - e^{-1.19907t}) + 1.07811 (-1 + e^{-1.19905t}))$$

221

0.37409

$$\left\{ \frac{-K[[5]] + \frac{-2s-2K[[6]]}{2s+4K[[7]]}}{K[[1]] \left(s + K[[2]] + \frac{s-K[[4]]}{2K[[3]]^2}\right)}, 7 \right\}$$

$$87641.5 e^{-1.19906t} (-1 + e^{-3.65022 \times 10^{-6}t}) - 394349. e^{-1.19906t} (-0.833988 (-1 + e^{1.19906t}) + 0.833986 e^{-3.65022 \times 10^{-6}t} (-1 + e^{1.19906t}))$$

Pozn.: Výsledky v příloze jsou ve formátu:

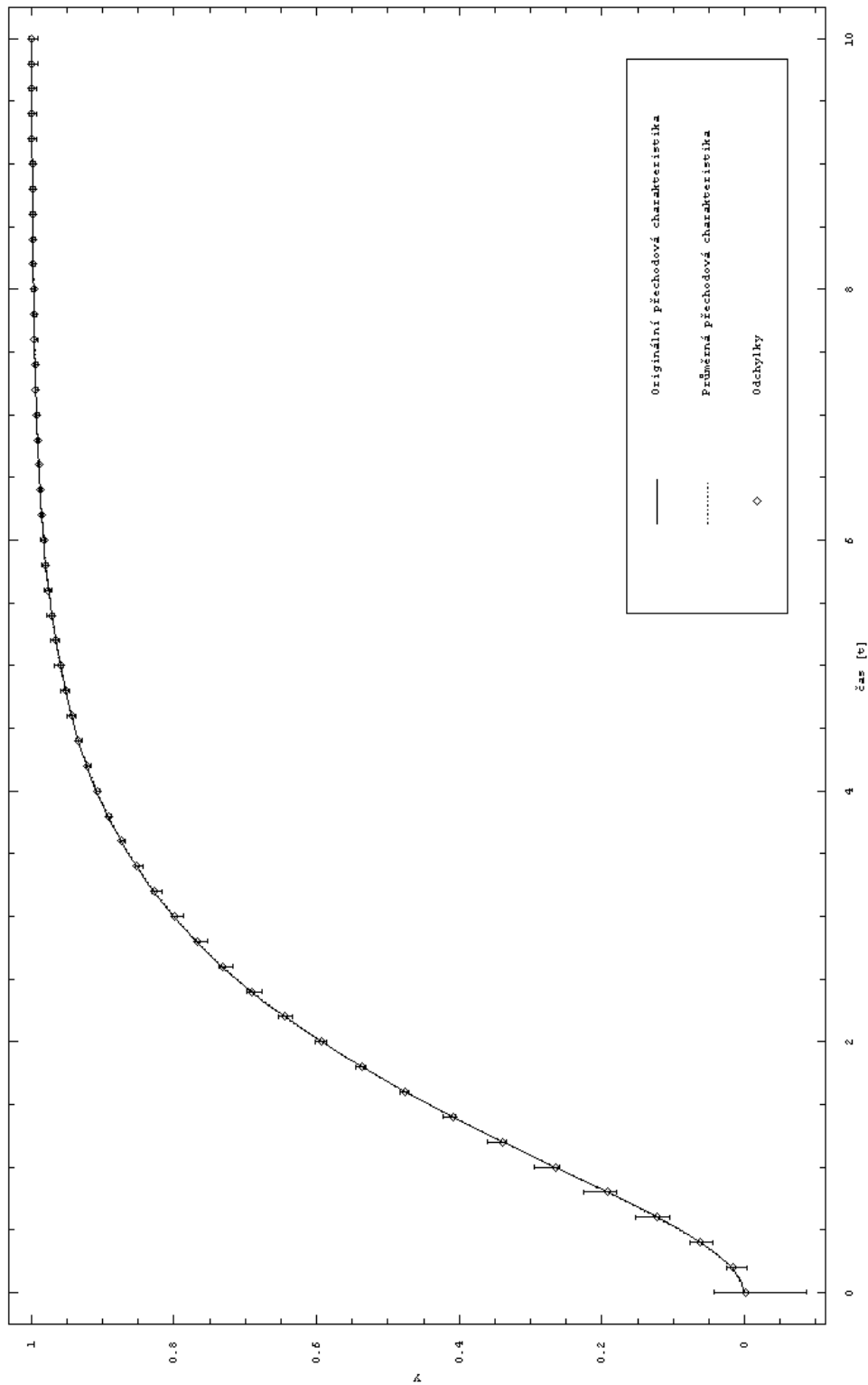
Počet ohodnocení účelové funkce

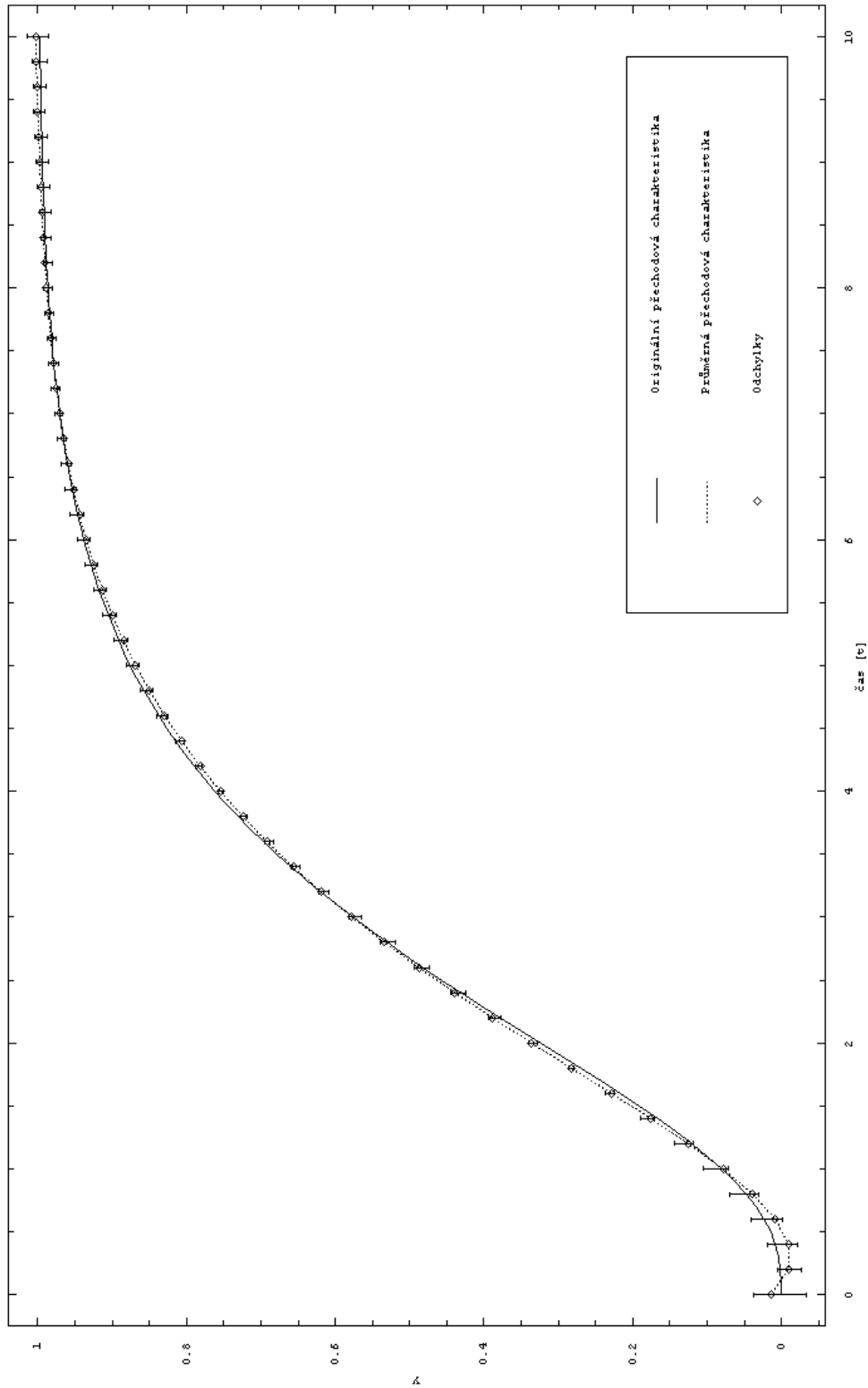
Hodnota účelové funkce

Nefitovaný tvar funkce

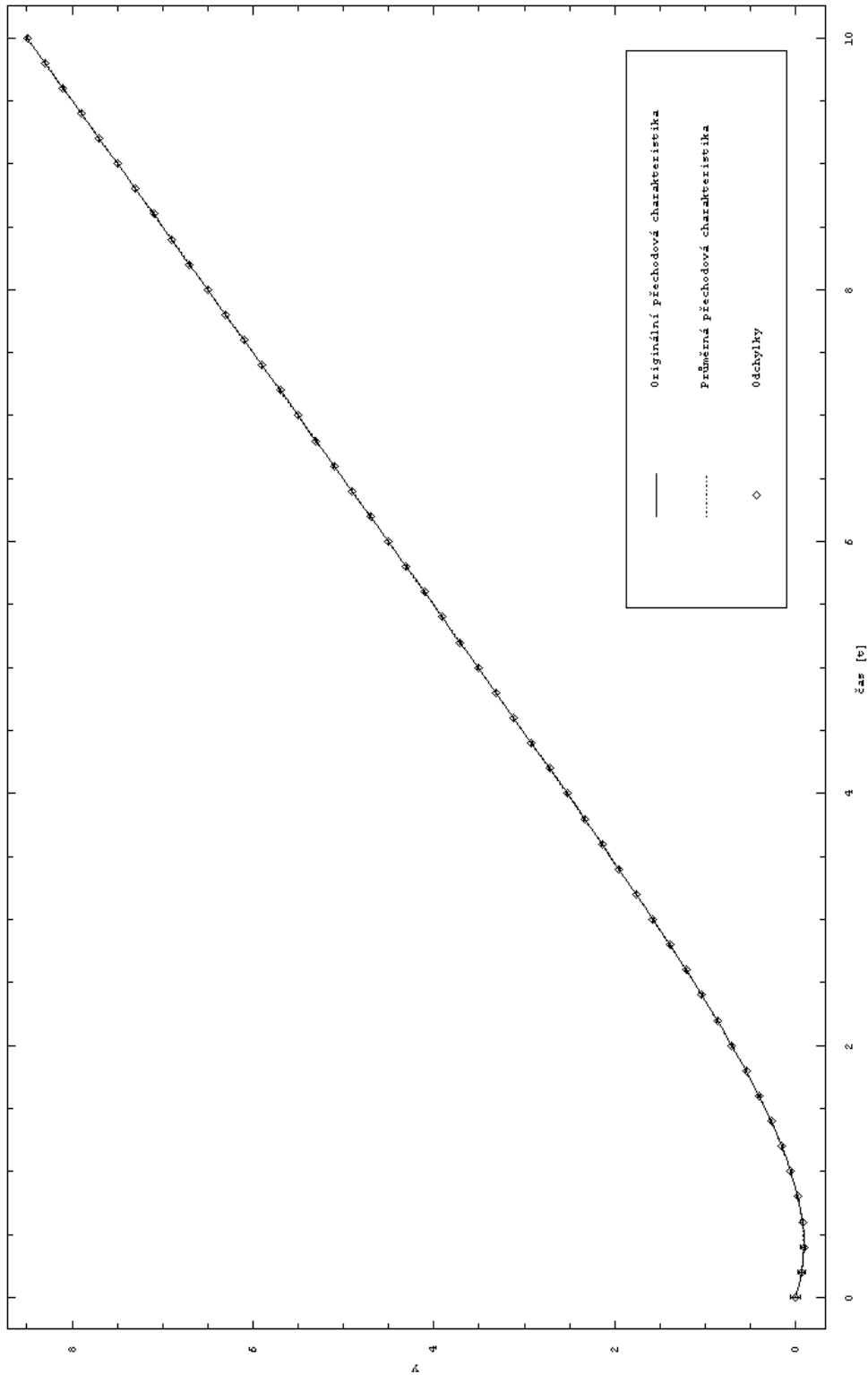
Fitovaný tvar funkce

PŘÍLOHA P VI: GRAFICKÝ PRŮBĚH PRO $\frac{1}{s^2 + 2s + 1}$



PŘÍLOHA P VII: GRAFICKÝ PRŮBĚH PRO $\frac{1}{(s+1)^3}$ 

PŘÍLOHA P VIII: GRAFICKÝ PRŮBĚH PRO $\frac{1-0,5s}{s(s+1)}$



PŘÍLOHA P IX: GRAFICKÝ PRŮBĚH PRO $\frac{1}{(1+s)(1+0,5s)(1+0,5^2s)(1+0,5^3s)}$

