

DETEKCE DISKREDIBILITY SENZORU U KOTLE NA BIOMASU OPTIMALIZAČNÍMI ALGORITMY

Ing. D. Klimánek, Doc. Ing. B. Šulc, Csc.*, Ing. J. Hrdlička***

*Ústav řídicí a přístrojové techniky, FS ČVUT v Praze, Technická 4, Praha, 166 07

**Ústav mechaniky tekutin a energetiky, FS ČVUT v Praze, Technická 4, Praha, 166 07

Abstrakt

Pro dosažení optimálního spalování paliva a snížení množství škodlivých produktů hoření lze u kotlů využít regulace přebytku vzduchu. U kotle na spalování biomasy, který je předmětem experimentálního výzkumu, je přebytek vzduchu při hoření zjišťován nepřímo pomocí kyslíkové sondy umístěné do odvodu spalin. Vzniká oprávněné nebezpečí, že pokud kyslíková sonda začne poskytovat zkreslené informace o přebytku kyslíku, je bez dalšího měření obtížné zjistit zhoršující se činnost kotle zejména zvýšenou produkcí škodlivin. Tato ztráta důvěryhodnosti senzoru se označuje jako diskredibilita senzoru. Příspěvek popisuje způsoby aplikace metod umělé inteligence (konkrétně standardní genetický algoritmus a algoritmus simulovaného žíhání), pomocí nichž lze, na základě určitého modelu senzoru, detekovat diskredibilitu senzoru, aniž by bylo nutné pořizovat další redundantní senzory.

1 Úvod

Pojmem diskredibilita senzoru je označován takový projev závad u senzoru, který nepředstavuje totální selhání senzoru, ale po němž již senzor přestává poskytovat správné informace o snímané veličině v rámci přípustných tolerancí. Tyto změny vlastností nejsou dobře rozlišitelné (např. u regulačních obvodů nebývá tato závada u senzoru regulované veličiny přímo těžko zjištělná), protože z hlediska vnějšího pozorování regulace probíhá bez chyby.

Jednou z možností detekce diskredibility senzoru je použití dalšího duplicitního senzoru, tzv. hardwarové redundance. Přestože tato hardwarová redundance může představovat dodatečné náklady, má své opodstatnění při řízení nebezpečných procesů. Příspěvek se pokouší ukázat novou, levnější cestu detekce diskredibility senzoru.

Aby se předešlo dodatečným nákladům na pořízení redundantního senzoru, je zkoumán způsob detekce diskredibility senzoru softwarovými nástroji, tj. využití tzv. softwarové redundance, která by měla nahradit funkci jednoho, nebo i více, často drahých dodatečných, měřicích zařízení. Cílem je vylepšit funkci regulátoru tak, aby indikoval změnu vlastností senzoru a signalizoval tuto skutečnost operátorovi. Pouze pokud je operátor včas informován o změnách funkce senzoru, může zabránit nebezpečným efektům plynoucím z nesprávné funkce senzoru a rozhodnout o výměně senzoru.

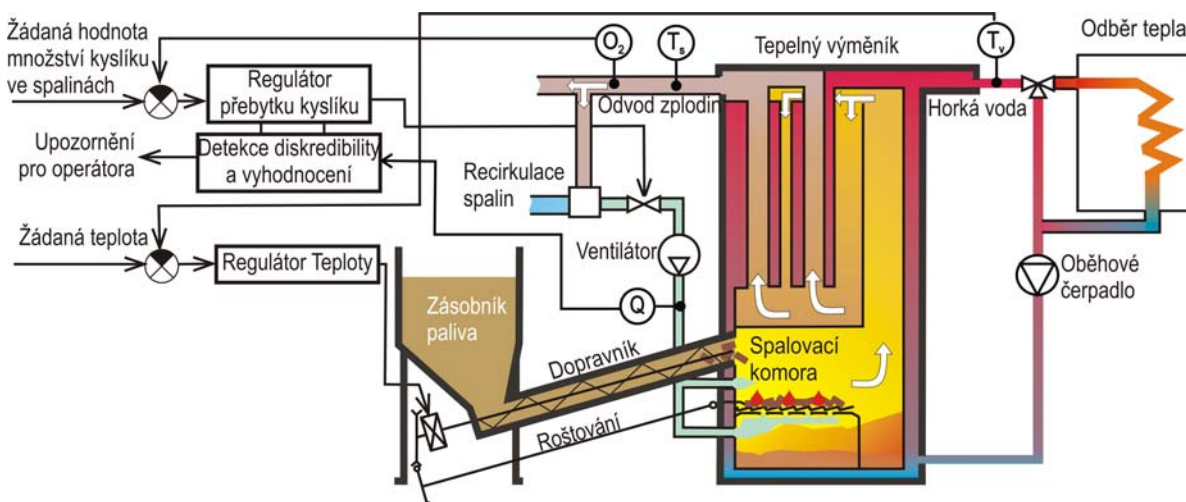
2 Detekce diskredibility senzoru

Na obr. 1 je schematicky znázorněn jednoduchý model kotle na spalování biomasy jako příklad zařízení, ve kterém je evidentní přínos detekce diskredibility senzoru. Součástí kotle je, z důvodu snížení množství emisí oxidů dusíku NO_x unikajících do ovzduší, zařízení sloužící k recirkulaci spalin. Při ní dochází ke smíchání vzduchu s částí spalin a výsledná směs je přiváděna do spalovací zóny. Z hlediska dosažení minimálních možných emisí CO_2 a NO_x , optimálního výkonu kotle a stability hoření biomasy je důležitá regulace přebytku vzduchu α ve směsi přiváděné do spalovací zóny (přičemž přebytek vzduchu α je poměr potřebného množství spalovacího vzduchu a teoretického vypočítaného množství). Hodnota α závisí na typu spalovacího zařízení a použitém palivu, v případě malého kotle a paliva z biomasy je hodnota α v rozmezí 1,8 až 2,2 [19].

Regulaci přebytku vzduchu α lze založit buď na ovládání přísunu biomasy do topeniště, nebo, jako v případě popisovaného experimentálního zařízení, na ovládání množství směsi vzduchu a spalin přiváděné do spalovací zóny. Přebytek vzduchu lze zjišťovat pomocí měření koncentrace kyslíku buď ve spalinách, nebo ve směsi vzduchu a spalin přiváděné do spalovací zóny. V popisovaném kotli je zjišťována koncentrace kyslíku ve spalinách, a to kyslíkovou sondou (λ -sondou) umístěnou do odvodu kotle. Přebytek vzduchu α souvisí s koncentrací kyslíku ve spalinách podle vztahu

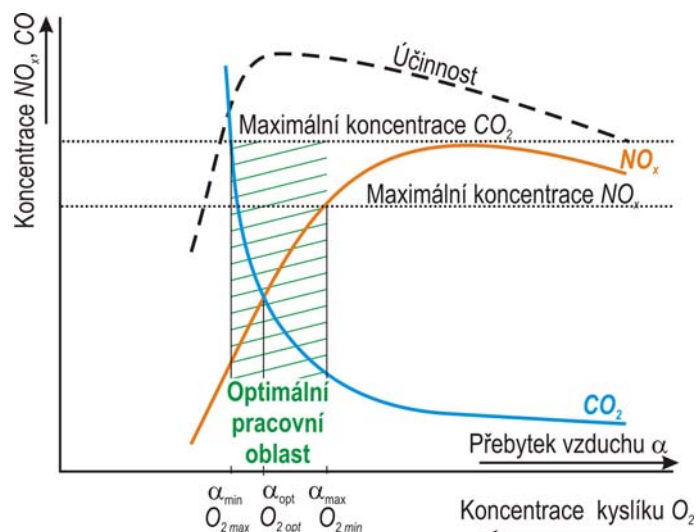
$$\alpha = \frac{20,9}{20,9 - O_2}, \quad (1)$$

kde O_2 je koncentrace kyslíku ve spalinách.



Obr. 1 Schéma kotle Verner pro spalování biomasy s detekcí diskredibility kyslíkové sondy

Obr. 2 zobrazuje optimální rozmezí přebytku vzduchu ve směsi přiváděné do spalovací zóny. Pokud se bude přebytek vzduchu α pohybovat v rozmezí α_{min} až α_{max} , pak hodnoty emisí CO_2 a NO_x spalinách nepřesáhnou maximální povolenou mez. Při dosažení optimálního přebytku vzduchu α_{opt} , bude zajištěno hoření biomasy s maximální účinností a minimem dosažitelných emisí CO_2 a NO_x , které jsou v případě spalování biomasy s neregulovaným přebytkem kyslíku obvykle vysoké.



Obr. 2 Koncentrace spalin v závislosti přebytku vzduchu ve spalinách

Důvod detekce diskredibility kyslíkové sondy je patrný z obr. 2. V případě nesprávně naměřených hodnot koncentrace kyslíku ve spalinách dojde vždy k nedokonalému hoření a zvýšení emisí CO_2 a NO_x , a to z důvodu nedostatku kyslíku ve spalovací zóně, nebo naopak z důvodu přebytku kyslíku s důsledkem přílišného ochlazení spalovací zóny. Bez dodatečných měření koncentrací škodlivých produktů hoření nelze rozpoznat zhoršující se hoření biomasy, protože z hlediska vnějšího pozorování probíhá regulace teploty horké vody bez chyby a zvýšené koncentrace škodlivých produktů hoření jsou pouze vedlejším efektem. Přesné analyzátoři plynu jsou drahé, a tak se provádí pouze kontrolní měření. Zjištěné nedostatky jsou sankcionovány za celé období od poslední kontroly. Proto je snaha využít kyslíkovou sondu nejenom ke zlepšení účinnosti a funkce kotle, ale také ke snížení emisí. K vyloučení (zmenšení) rizika zvýšené koncentrace škodlivých produktů hoření se obvykle používá redundantní kyslíková sonda. Ekonomičtější řešením může být použití počítačového modelu a programu pro zpracování měřených dat, tzv. softwarové redundance.

3 Detekce diskredibility senzoru evolučními algoritmy

Pro detekci diskredibility senzoru není známa aplikace metod používaných pro detekci poruchových stavů systému, např. kvantitativní modelové metody, kvalitativní modelové metody nebo metody založené na historii procesních dat [1]. Metoda detekce diskredibility senzoru je založena na modelu senzoru, který lze formulovat jako funkci jiných, někdy i nesnímaných veličin, tzv. parametrů modelu. Hlavní výhodou navrhovaného řešení je, že nezbytné informace o změnách vlastností senzoru mohou být získány na základě provozních dat, která jsou zaznamenávána během provozu zařízení. Pro detekci diskredibility senzoru je potřeba stanovit odchylku $e(t)$ jako rozdíl výstupu modelu senzoru od výstupu skutečného senzoru

$$e(t) = |y_{est}(t) - y_{real}(t)|, \quad (2)$$

kde $y_{est}(t)$ je estimovaný výstup modelu senzoru (v případě detekce kyslíkové sondy y_{est} představuje předpokládanou koncentraci kyslíku ve spalínách získanou teoretickým výpočtem) a $y_{real}(t)$ je výstup skutečného senzoru (naměřená hodnota koncentrace kyslíku ve spalínách). Svou velikostí odchylka $e(t)$, obvykle nazývaná reziduum, indukuje míru nesouladu mezi estimovaným výstupem modelu senzoru a údaji ze senzoru.

Pokud má model vyjadřovat statické chování senzoru, pak lze obecný model senzoru vyjádřit skalární funkcí

$$y_{est}(t) = f(\mathbf{p}, \mathbf{x}), \quad (3)$$

kde \mathbf{x} je vektor odhadů vstupů senzoru a \mathbf{p} představuje vektor n parametrů modelu senzoru. Algoritmus detekce diskredibility senzoru je založen na hledání optimálního vektoru parametrů \mathbf{p} modelu senzoru tak, aby reziduum bylo minimální, tzn. rovno nebo alespoň blízké nule. Optimální vektor parametrů \mathbf{p} modelu senzoru se průběžně vyhodnocuje. Pokud nedojde k diskredibilitě senzoru, vektor optimálních parametrů \mathbf{p} je stále stejný, zatímco změna některého z parametrů může signalizovat počáteční diskredibilitu senzoru. Tím byl problém detekce diskredibility převeden na sledování trendu jednotlivých parametrů modelu senzoru s uplatněním metod optimalizačních algoritmů.

Pro nalezení optimálního vektoru parametrů modelu senzoru, neboli minimalizaci rezidua, lze použít jakoukoli numerickou vícerozměrovou statickou optimalizační metodu (např. gradientní metodu, Lagrangeovu metodu, optimalizační evoluční algoritmy). Problém je, že vstupem do senzoru je neznámá, dynamicky se měnící veličina. Proto by pro hledání vektoru parametrů modelu senzoru bylo vhodné použít metodu, která by při hledání optimálního vektoru parametrů modelu senzoru zohlednila také prvek náhodného výběru. Tento požadavek splňují optimalizační metody umělé inteligence, pro které byl zaveden termín evoluční algoritmy (EA) [2]. Uváděná časová náročnost EA při hledání optimálního vektoru parametrů \mathbf{p} modelu senzoru nevádí, protože z charakteru vývoje diskredibility senzoru vyplývá, že ztráta kredibility senzoru má dlouho trvající vývoj, který je v porovnání s dobou hledání vektoru parametrů modelu senzoru mnohem delší.

4 Evoluční algoritmy

EA jsou netradiční výpočetní postupy, které nacházejí svou inspiraci v přírodních vývojových procesech. Metodám EA a jejich aplikacím se věnuje mnoho publikací, které však obvykle používají terminologii přeboranou z popisu přírodních procesů. Tento příspěvek se snaží používané termíny EA převést na termíny spojené s detekcí diskredibility senzorů.

Stejně jako přírodní jevy, mají i EA svou náhodnou složku, která nezanedbatelným způsobem ovlivňuje jejich celkové vlastnosti a chování [3]. Nová řešení vznikají náhodnými změnami předchozích řešení a pokud jsou nová řešení výhodnější, nahrazují předchozí řešení. Volba metody pro nalezení optimálního vektoru parametrů modelu senzoru pomocí EA vyžaduje rovnováhu dvou cílů:

- co nejrychleji najít nejbližší vektor parametrů modelu senzoru, při kterém je dosaženo pouze lokálního rezidua, v malém okolí náhodně zvoleného výchozího vektoru,
- co nejlépe prohledat prostor všech přípustných hodnot parametrů a najít optimální vektor parametrů modelu senzoru.

Metody EA se liší mírou uplatnění uvedených strategií a je možné je seřadit podle této míry od metod, které naleznou nejbližší lokální optimum, až k metodám prohledávající celý prostor řešení. Toto seřazení je následované [9]:

- horolezecký algoritmus,
- zakázané prohledávání,
- simulované žíhání

- evoluční strategie,
- standardní genetický algoritmus

Pro účely detekce diskredibility senzoru pomocí EA byla rozpracována a testována metoda standardního genetického algoritmu a metoda simulovaného žihání.

Náhodný výběr počátečního vektoru parametrů modelu senzoru a ohodnocení vektorů parametrů modelu senzoru

Všechny EA vychází z počátečního řešení, v případě detekce diskredibility senzoru jím je vektor parametrů modelu senzoru (dále jen vektor parametrů). Počáteční vektor parametrů p (v případě standardního genetického algoritmu skupina vektorů parametrů) je inicializován náhodně, s normálním rozložením a v rozsahu přípustných hodnot jednotlivých parametrů.

Dále je pro EA společný mechanismus posuzování vhodnosti vektoru parametrů pomocí hodnoty rezidua. V případě standardního genetického algoritmu se pro ohodnocení používá pojem fitness. Ohodnocení vektorů parametrů slouží k určení výběru vektorů pro další iteraci.

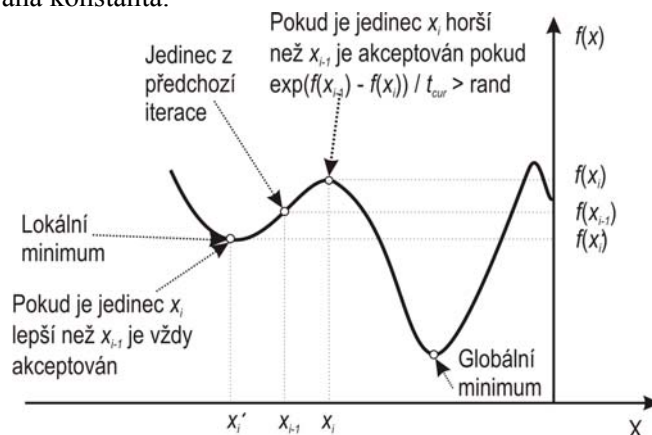
4.1 Metoda simulovaného žihání

Simulované žihání (*Simulated Annealing*) (SA) je metoda, která pro akceptaci nového řešení využívá náhody. Název metody je odvozen z představy simulování fyzikálních procesů probíhajících při odstraňování defektů krystalové mřížky kovu, které se projevují pnutím v materiálu. Při žihání se kov zahřeje na tak vysokou teplotu, při které atomy v krystalové mřížce mohou překonat lokální energetické hladiny, a defekty krystalové mřížky mají velkou pravděpodobnost zániku. Po dosažení tohoto stavu se kov pomalu ochlazuje (žihá), a tím se atomy dostanou do rovnovážných poloh s nejmenší energií [11]. Při konečné teplotě žihání (podstatně nižší, než byla počáteční) jsou všechny atomy kovu v rovnovážných polohách a těleso neobsahuje žádné vnitřní defekty ani pnutí.

V simulovaném žihání je krystal reprezentován jedincem x . Ke každému jedinci může být přiřazena funkční hodnota $f(x)$, která představuje energii krystalu. Analogií minimalizace energie krystalu je v metodě simulovaného žihání minimalizace funkce $f(x)$ a pomalé ochlazování představují postupné iterace. V každé iteraci je původní jedinec x_{i-1} nahrazen novým, náhodně vygenerovaným jedincem x_i . Pravděpodobnost nahrazení je

$$P(x_{i-1} \rightarrow x_i) = \begin{cases} 1, & \text{if } f(x_{i-1}) \leq f(x_i) \\ e^{-\frac{f(x_{i-1}) - f(x_i)}{t_{cur}}}, & \text{otherwise} \end{cases} \quad (4)$$

kde t_{cur} je parametr vyjadřující teplotu v daném kroku. Jestliže jedinec x_i má menší, nebo stejnou funkční hodnotu jako původní jedinec x_{i-1} , je automaticky akceptován do další iterace. V opačném případě je pravděpodobnost akceptování jedince x_i menší než jednotková, ale i v tomto případě má nový jedinec šanci postoupit do další iterace, viz obr. 3, kde $rand$ je náhodně, s normálním rozložením, vygenerovaná konstanta.



Obr. 3 Princip hledání minima funkce $f(x)$

Metodu SA znázorňuje pseudokód SA na obr. 4. Řídicí parametr t_{cur} je ohraničen maximální a minimální hodnotou, $t_{min} < t_{cur} < t_{max}$, snižování teploty je realizováno vztahem $t_{cur} = \lambda \cdot t_{cur}$, kde λ je kladné číslo menší než jedna, obvykle $\lambda = 0,9$. Proměnná t zaznamenává celkový počet "pokusů" simulovaného žihání pro daný řídicí parametr t_{cur} . Pro volbu konstant t_{max} a t_{min} neexistuje všeobecný

předpis, jejich určení závisí na zkušenostech. Reálná proměnná *rand* je náhodně generované číslo (s normálním rozdělením) z intervalu $\langle 0, 1 \rangle$. Jedinec \mathbf{x}^* zaznamenává nejlepší řešení v průběhu celého simulovaného žihání. Ve všeobecnosti jedinec \mathbf{x} po skončení simulovaného žihání nemusí být rovný jedinci \mathbf{x}^* .

```

t = 0; inicializace náhodného jedince x(t);
tcur = tmax; x* = x(t);
ohodnocení jedince f(x(t));
while (tcur > tmin) do
  begin t = t + 1;
    generování náhodného jedince x(t);
    ohodnocení jedince f(x(t));
    if f(x(t)) ≤ f(x(t-1)) THEN Pr = 1 ELSE
      P = exp( - (f(x(t)) - f(x(t-1))) / tcur );
    if rand < Pr THEN
      begin x = x(t-1);
        if f(x) < f(x*) then x* := x;
        end;
      end;
      tcur = λ tcur;
    end;
  end;

```

Obr. 4 Obecný pseudokód algoritmu simulovaného žihání

Implementace SA pro nalezení optimálního vektoru parametrů s přihlédnutím na realizaci v programu Matlab

Cílem je obecný popis principu SA vysvětlit v pojmech užívaných pro hledání optimálního vektoru parametrů a aplikovat na řešení úlohy, při kterém bude uvažována realizace v programu Matlab.

V analogii o fyzikální představě žihání představuje krystal vektor parametrů modelu \mathbf{p} a energie krystalu odpovídá hodnotě rezidua $z = e(\mathbf{p})$. Pro nalezení optimálního vektoru parametrů s pomocí algoritmu SA v obr. 5 jsou vyžadovány následující kroky:

1. Při inicializaci algoritmu se provede nastavení počáteční a konečné hodnoty parametrů t_{max} a t_{min} , a řídicího parametru $t_{cur} = t_{max}$. Tyto parametry představují formální analogii teploty a jsou používány k testu zastavení cyklu hledání řešení. Řídicí parametr t_{cur} , je použit při výpočtu Boltzmanova kritéria, které rozhoduje o akceptaci nového vektoru parametrů v kroku 4. Pořadí iterace i je nastaveno na hodnotu jedna a je inicializován váhový koeficient λ rozsahu $(0-1)$.
2. V rozsahu předpokládaných hodnot parametrů je náhodně inicializován vektor parametrů $\mathbf{p}(i)$ a je zjištěno reziduum $z(1)$, pořadí iterace i je zvýšeno o jedna.
3. Stejně jako v kroku 2 je vygenerován náhodný vektor $\mathbf{p}(i)$ a je určena odpovídající hodnota kritéria kvality nesouladu $z(i)$.
4. Je učena diference mezi rezidui v po sobě jdoucích iteracích $\Delta = z(i) - z(i - 1)$. Pokud je $\Delta < 0$, pak je vektor řešení $\mathbf{p}(i)$ automaticky akceptován. V opačném případě je v intervalu $\langle 0-1 \rangle$ vygenerována proměnná *rand*. Nový vektor parametrů může být akceptován na základě porovnání Boltzmannova kritéria(4), tj. pokud

$$\exp(-\Delta / t_{cur}) > rand. \quad (5)$$

5. Řídicí parametr t_{cur} je v každé iteraci násoben váhovým koeficientem λ ,
6. Hledání vektorů parametrů je zastaveno, pokud je t_{cur} menší než t_{final} . Jinak návrat do kroku 3.

```

inicializovat řídící parametry  $t_{cur} = t_{max}, t_{final}$ , pořadí iterace  $i$  nastavit
na hodnotu jedna, inicializovat váhový koeficient  $\lambda$ 
inicializovat vektor potenciálních parametrů senzoru  $\mathbf{p}(1)$ 
určit reziduum  $z(1)$ 
zvýšit číslo iterace  $i = i + 1$ 
if  $t_{cur} > t_{final}$ 
    náhodně zvolit nový vektor potenciálních parametrů senzoru  $\mathbf{p}(i)$ 
    určit reziduum  $z(i)$ 
    stanovit Boltzmannovo kritérium  $P = \exp(z(i) - z(i - 1)) / t_{cur}$ 
    if  $z(i - 1) < z(i)$  or  $P < rand$ 
        akceptovat nový vektor parametrů
    else
        návrat k původním hodnotám rezidua  $z(i) = z(i - 1)$ .
    end
    zvýšit pořadí iterace  $i = i + 1$ 
    snížit řídící parametr  $t_{cur} = \lambda t_{cur}$ ;
end

```

Obr. 5 Pseudokód algoritmu simulovaného žíhání sesatavený pro nalezení optimálního vektoru parametrů \mathbf{p}

4.2 Standardní genetický algoritmus pro hledání optimálního vektoru parametrů modelu senzoru

Genetické algoritmy (*Genetic Algorithms*) (GA) vycházejí z Darwinovy teorie o vývoji druhů. Zjednodušená biologická interpretace objasňuje GA následovně. Všechny živé organismy jsou složeny z buněk. Každá z buněk obsahuje jádro, které v rámci jednoho organismu obsahuje identické kopie chromosomu. Chromosomy se dále dělí na geny (podmnožiny chromosomu) popisující jednu vlastnost organismu. Při rozmnožování se páry chromosomu jednoho jedince zkříží a vytvoří tak nový chromosom. Nově vzniklý chromosom se podrobí mutaci, kdy se náhodně změní hodnoty genu. Schopnost organismu přežít v daném prostředí se označuje fitness.

Popsaný mechanismus probíhá podle zákonů, které se GA snaží napodobovat. Způsoby tohoto provedení se mohou lišit. Interpretace GA v tomto příspěvku se zabývá popisem nejčastěji se vyskytující verze GA, která se nazývá standardní genetický algoritmus.

V kontextu standardního genetického algoritmu (SGA) představuje chromosom jedno přípustné řešení daného problému, které označíme x . Existují dva způsoby kódování chromozomu – pracuje se s dekadickým, nebo binárním chromozomem. Následující popis se věnuje pouze binárnímu kódování chromozomu, který je reprezentovaný souborem dvouhodnotových proměnných. Jedinci jsou kódováni v předem stanoveném počtu bitů. Počet bitů L závisí na zvolené přesnosti řešení ε . Hledáme-li extrém funkce $f(x)$ na intervalu $\langle a, b \rangle$ s přesností lepší než ε , pak je potřebná délka L binárního chromozomu

$$L = \left\lceil \log_2 \frac{|b - a|}{\varepsilon} \right\rceil \quad (6)$$

Bude-li každé x reprezentováno binárním L -bitovým řetězcem, je každý řetězec popsán

$$\left(\langle b_{L-1}, b_L, \dots, b_1, b_0 \rangle \right)_2 = \left(\sum_{i=0}^{L-1} b_i 2^i \right)_{10} = x' \quad (7)$$

Řetězec x' reprezentuje dekadickou hodnotu

$$x = a + x' \frac{|b - a|}{2^L - 1} \quad (8)$$

Vztah (7) představuje kódování jedince a vztah (8) představuje dekódování jedince. V terminologii obecné genetiky je řetězec x' analogií chromozomu kódujícího genetickou informaci (genotyp). Navenek se pak tato genetická informace zobrazí do konkrétního jedince (fenotyp). Číslo x lze tak považovat za fenotypový obraz chromozomu [7].

GA pracuje s množinou chromosomů a tato množina se nazývá populace. Velikost populace n_p udává počet jejích chromozomů. Vývojový čas SGA běží v diskrétních krocích a v tomto časovém

pohledu se hovoří o posloupnosti populací jako o generacích.

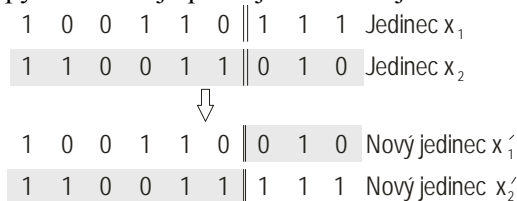
Mechanismus přirozeného výběru v přírodě, při kterém přežívají jen nejsilnější jedinci, je realizován operátorem selekce. Účelem selekce je vybrat do další generace pouze kvalitní jedince (odfiltrování horších jedinců v příští generaci). Existuje mnoho různých modelů selekce, nejpoužívanější je metoda ruletového kola. Jinou možností selekce je např. prosté vybrání N nejlepších a M nejhorších jedinců, výběr jedinců, jejichž zdatnost je lepší než nějaká mez a výběr těch, jejichž zdatnost je horší než nějaká mez, nebo turnajový mechanismus, ve kterém vedle sebe stojící jedinci mezi sebou bojují o přežití [15]. Při selekci pomocí metody ruletového kola jednotlivé chromosomy získají na obvodu ruletového kola takovou část, která odpovídá poměru jejich fitness k průměru fitness populace. Pravděpodobnost P_r výběru jedince do další generace je dána vztahem

$$P_r(i) = \frac{f(x_i)}{\sum_{j=1}^{n_p} f(x_j)}, i = 1, 2, \dots, n_p \quad (9)$$

kde $f(x_i)$ je fitness každého jedince a n_p velikost populace. Mechanismus ruletového kola vychází z představy, že se "točí kolem" a je vybírán ten jedinec, na jehož úseku se kolo zastaví. Stejněho jedince je možno zvolit vícekrát. Je zřejmé, že čím má chromosom vyšší hodnotu fitness, tím větší má šanci být vybrán do nové generace. Selekcí vznikne nová populace.

Rozmnožování jedinců je simulováno změnovými operátory křížení a mutace, které jsou aplikovány na jedince vybrané selekcí. Ne všichni vybraní jedinci však musí těmito genetickými operacemi projít.

Proces křížení je simulován tak, že se nad každým selektovaným jedincem provede náhodný pokus a příslušný jedinec se s pravděpodobností P_{cross} stane kandidátem na reprodukci. Vznikne tak množina jedinců, z nichž se vybírají dvojice buď náhodně, nebo se za pár prohlásí po sobě jdoucí jedinci tak, jak byli do této množiny zařazeni. Je-li počet jedinců lichý, jeden kandidát se vyloučí. Bod křížení je také obvykle náhodný. Jednou z možností křížení je binární jednobodové křížení zobrazené na obr. 6. Existují např. postupy křížení nejlepších jedinců s nejslabšími a vícebodové křížení.



Obr. 6 Příklad binárního jednobodového křížení

Po operátoru křížení následuje simulace biologické mutace a jejím hlavním úkolem je zajistit, aby populace nedegradovala v identické chromosomy. Mutace náhodně změní vybraný bit chromosomu z 0 na 1 a naopak. Pravděpodobnost mutace musí být nenulová, aby byly zaručeny změny, ale nesmí být příliš vysoká, aby byl zachován prvek dědičnosti.

Obecný tvar algoritmu SGA představuje obr. 7. Algoritmus běží v cyklu, ve kterém t simuluje vývojový čas. Algoritmus běží tak dlouho, dokud není splněna podmínka zastavovacího pravidla. Dobrá formulace zastavovacího pravidla je důležitá, protože ovlivňuje dobu hledání řešení. Volba zastavovacího pravidla není jednoduchá a není známé žádné optimální kritérium. Možným zastavovacím pravidlem je např. ukončení algoritmu ve chvíli, kdy hodnota fitness nejlepšího jedince, nebo průměrná fitness celé populace bude rovna předem stanovené hodnotě, nebo omezení počtu generací, které může algoritmus použít.

```

t = 0;
inicializace počáteční populace  $\mathbf{x}(t)$ ;
určení fitness populace  $\mathbf{x}(t)$ ;
while not zastavovací_pravidlo do
  begin t = t+1;
  vyber novou populaci  $\mathbf{x}(t)$  z předchozí populace  $\mathbf{x}(t-1)$ ;
  aplikace genetických operátorů mutace a křížení na populaci  $\mathbf{x}(t)$ ;
  aplikací genetických operátorů vznikne modifikovaná generace  $\mathbf{x}(t)$ ;
  určení fitness populace  $\mathbf{x}(t)$ ;
  end;
end;
```

Obr. 7 Pseudokód standardního genetického algoritmu

Pseudokód v obr. 7 může být popsán následovně.

1. Inicializace počáteční populace

Před prvním výpočetním cyklem se náhodně inicializuje počáteční populace $\mathbf{x}(t)$ možných řešení

$$\mathbf{x}(t) = \{x_i(0), i = 1, 2, \dots, n_p\}, \quad (10)$$

kde n_p představuje velikost populace a t představuje vývojový čas.

2. Určení fitness populace

Příhodnost každého jedince pro další generaci se ohodnotí pomocí fitness $\Phi(p_i(t))$

$$\Phi(\mathbf{x}(t)) = \{\Phi(x_i(t)), i = 1, 2, \dots, n_p\}, \quad (11)$$

kde n_p představuje velikost populace.

3. Vytvoření nové populace

Vývojový čas se zvýší o jedna a pomocí selekce se z populace $\mathbf{p}(t-1)$ vyberou jedinci pro novou populaci

$$\mathbf{x}'(t) = \{s_{ns}(x_i(t-1)), i = 1, 2, \dots, n_p\}, \quad (12)$$

kde s_{ns} představuje zvolenu metodu selekce.

4. Křížení a mutace

Vybraná populace podstoupí křížení a mutaci s předem určenou pravděpodobností P_{cros} a P_{mut} v rozsahu $<0-1>$

$$\mathbf{x}''(t) = \{rP_{cros}(x'_i(t)), i = 1, 2, \dots, n_p\}, \quad (13)$$

kde r představuje způsob křížení s pravděpodobností křížení P_{cros}

$$\mathbf{x}(t) = \{P_{mut}(x''_i(t)), i = 1, 2, \dots, n_p\}, \quad (14)$$

kde P_{mut} představuje pravděpodobnost mutace.

5. Pokud je splněná podmínka ukončovacího kritéria, skok zpět do bodu 2.

Implementace SGA pro nalezení optimálního vektoru parametrů s přihlédnutím na realizaci v programu Matlab

Předvedený obecný popis SGA je vhodné převést do termínů, které jsou spojeny s hledáním optimálního vektoru parametrů pro detekci diskredibility senzoru. V analogii popsaného SGA představuje populace skupinu vektorů parametrů, chromozomy představují jednotlivé vektory parametrů.

Pro práci se standardním genetickým algoritmem existuje mnoho programovacích prostředků, např. programy Genesis, Gentior, Genetic Algorithm Toolbox, atd. Pro sestavení popsaného SGA pro nalezení optimálního vektoru v programu Matlab/Simulink byl použit Genetic Algorithm Toolbox [15], protože je navržen přímo pro prostředí Matlab.

Při sestavení SGA je nutno vzít v úvahu střídání generací a běh simulace tak, aby bylo možné stanovit fitness (hodnotu rezidua) vektorů parametrů. Pseudokód algoritmu SGA vhodného nalezení optimálního vektoru parametrů na obr. 8 lze popsat následujícími kroky:

1. Při inicializaci SGA je evoluční čas nastaven na nulu ($t = 0$), v rozsahu předpokládaných hodnot je inicializována počáteční skupina potenciálních vektorů parametrů $\mathbf{p}(t)$, je nastaveno zastavovací pravidlo – minimální požadovaná průměrná hodnota rezidua celé skupiny parametrů vektorů a nastaven parametr mutace P_{mut} a křížení P_{cros} .
2. Ke každému vektoru parametrů je přiřazena hodnota rezidua. Dokud není průměrná hodnota rezidua menší, než požadovaná, je na skupinu vektorů postupně aplikován operátor selekce, křížení, mutace.
3. Pro selekci vektorů Genetic Algorithm Toolbox nabízí metody selekce (metoda ruletového kola, Stochastic Universal Sampling). Metody předpokládají hledání maxima funkce na základě rezidua vektorů parametrů. Proto je nutné zařadit před selekci funkci, která přehodnotí velikost rezidua tak, že nejmenším hodnotám reziduí přiřadí největší hodnotu a naopak. Tato operace se nazývá rankling a je aplikována použitím funkce *rankling* [15]. Funkce *rankling* ohodnotí vektor reziduí tak, aby nejlepší vektor parametrů získal fitness dva a nejhorší vektor parametrů hodnotu fitness nula.
4. Na selekci vybrané vektory parametrů jsou aplikovány změnové operátory. Ne všechny dvojice podstoupí křížení, o křížení rozhodne pravděpodobností křížení P_{cros} . Pro realizaci binárního jednobodového křížení po sobě jdoucích dvojic parametrů je nutné převést jednotlivé chromosomy do binárních řetězců. Ke každé dvojici chromosomů se funkcí *rand* náhodně vygeneruje proměnná

Rand v intervalu $\langle 0-1 \rangle$ představující, kolik procent binárního řetězce prvního chromosomu zůstane zachováno. Ke křížení nedojde, pokud $P_{cros} > Rand$.

Pro realizaci mutace je nutné nad každým prvkem binárního chromosomu náhodně vygenerovat proměnnou *Mut*. K inverzi daného prvku dojde, pokud je $P_{mut} > Mut$. Nově vzniklou binární populaci je nutno převést do dekadické soustavy, aby bylo možné určit velikost rezidua vektorů parametrů. Rozšířením SGA implementovaným do algoritmu je tzv. elitismus, zachování nejlepšího vektoru do příští generace. Vektor s nejlepším ohodnocením je automaticky překopírován do následující generace beze změny. Návrat do bodu 3.

5. Pokud je SGA ukončen, je ze skupiny vektorů parametrů $\mathbf{p}(t)$ vybrán optimální vektor parametrů na základě velikosti reziduí.

inicializovat pravděpodobnosti mutace a křížení, evoluční čas nastavit na nulu.

inicializovat počáteční populaci (skupinu parametrů vektorů) $\mathbf{p}(t)$;

stanovit hodnotu rezidua pro každý vektor ze skupiny vektorů $\mathbf{p}(t)$;

pokud není splněno zastavovací pravidlo

$t = t+1$;

aplikovat funkci *rankling* pro přehodnocení velikosti reziduí

selektce nové populace $\mathbf{p}(t)$ z předchozí populace $\mathbf{p}(t-1)$;

aplikovat operátor křížení a mutace na populaci $\mathbf{p}(t)$;

aplikací genetických operátorů vznikne nová populace $\mathbf{p}(t)$;

zjistit velikosti rezidua pro každý vektor ze skupiny vektorů $\mathbf{p}(t)$;

end;

výběr nejlepšího vektoru parametrů z populace $\mathbf{p}(t)$;

Obr. 8 Pseudokód standardního genetického algoritmu sestaveného pro nalezení optimálního mentoru

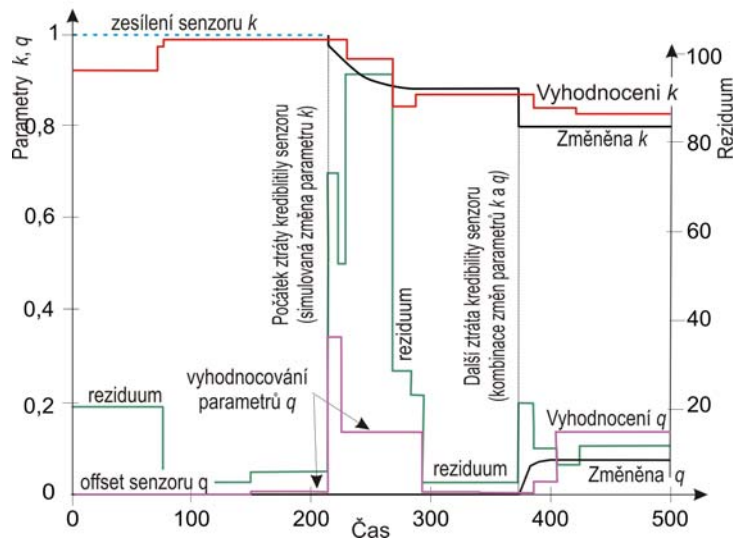
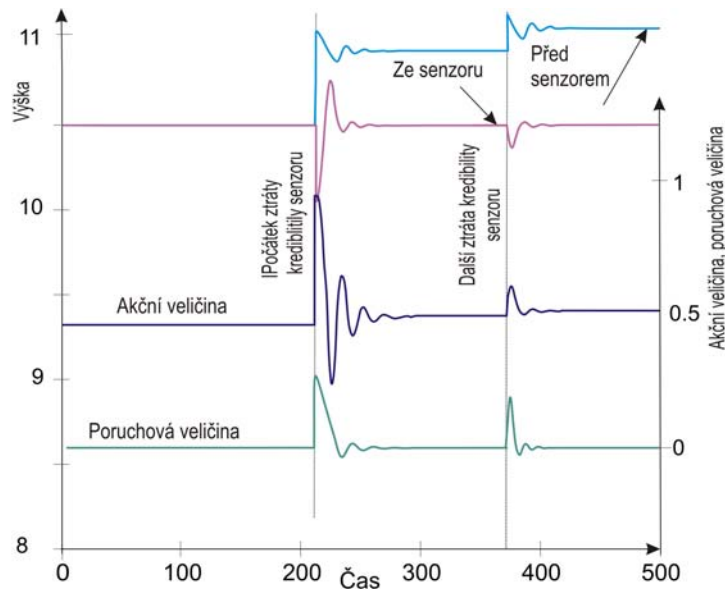
Obr. 9 představuje výpis části kódu M-file programu Matlab pro vytvoření nové generace použitého v bloku SGA. Kód obsahuje funkce Matlab Algorithm Toolbox. Z důvodu hledání minima je před selekci zařazena funkce *rankling*. Metody křížení a selektce pracují s binární prezentací parametrů vektorů.

```
% SGA rekombinační operace
FitnV = ranking(ObjV); % přehodnocení velikosti reziduí
SelCh = select('rws', Pop, FitnV); %selektce metodou ruletového kola
BIN=f2b(SelCh, bounds, bits); %převod do binární podoby
XSelChBIN=Xover((BIN),px); % jednobodové binární křížení
MSelChBIN=mutation(XSelChBIN,pmut); % mutace
SelCh =b2f(MSelChBIN, bounds, bits); %převod nové populace do dekadické
soustavy.
```

Obr. 9 Výpis části kódu M-file programu Matlab znázorňující řazení rekombinačních operátorů pro tvorbu nové generace

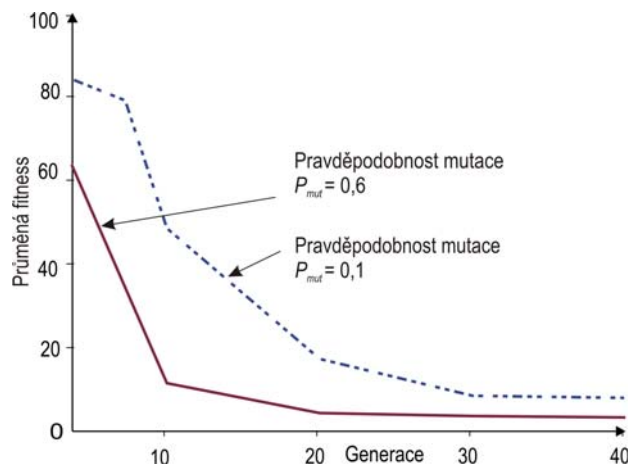
5 Závěry z ověřování použitelnosti popisovaných metod

Pro získání a porovnání srovnatelných výsledků bylo vhodné vyjít ze simulace činnosti jednoduchého regulačního obvodu. Chování a zákonitosti změn regulované veličiny, jejíž velikost lze jednoznačně deterministicky určit, snímá sensor, jehož diskredibilita je předmětem zájmu. Tento jednoduchý regulační obvod představoval model kaskády dvou nádrží vytvořený v programu Matlab/Simulink, který respektoval nelinearity a ošetření mezních hodnot otevření ventilů, velikosti akční veličiny atd. Pro ověřování použitelnosti metod SGA a SA je významnou částí modelu lineární model senzoru výšky hladiny, který má dva parametry – zesílení a tzv. offset. Pro tento model jsou estimované hodnoty modelu vstupu senzoru přesně zjistitelné na základě deterministických vztahů a vazeb. Simulační experimenty probíhaly tak, že v modelu byla simulována změna (obvykle skoková) jednoho nebo obou parametrů senzoru s možností zvolit metodu SGA nebo SA pro ověření jejich schopnosti detekovat diskredibilitu. Typická ukázka detekce diskredibility senzoru pro metodu SGA je na obr. 10, obdobný průběh detekce byl zaznamenán i u metody SA. Provedené experimenty prokázaly, že metody SGA a SA jsou vhodnými nástroji při detekci diskredibility senzoru, protože při simulaci byl patrný trend vývoje parametrů modelu senzoru, a to i v případě přechodových stavů modelu. Z porovnání metod SA a SGA plyne, že výsledky v závislosti na metodě se zásadně neliší, u obou metod je důležité optimální nastavení jejich parametrů.



Obr. 10 Důsledek ztráty kredibility senzoru a průběh vývoje parametrů při minimalizaci rezidua pomocí SGA

Při testování optimálního nastavení parametrů metody SGA se projevilo, že příliš velká pravděpodobnost mutace nevede k příznivému vývoji vektorů parametrů a příliš malá pravděpodobnost mutace nepřináší dostatek změn do nové populace. Na obr. 11 je znázorněna velikost průměrné hodnoty rezidua v jednotlivých generacích na pravděpodobnosti mutace.



Obr. 11 Vliv pravděpodobnosti mutace na průměrnou hodnotu rezidua.

V testovaném případě byla optimální velikost populace $n_p = 30$ a parametr křížení $P_{cros} = 0,6$. Při testování metody SA byly nejlepší výsledky dosaženy při volbě váhového koeficientu $\lambda = 0,8$ a řídicích parametrů $t_{max} = 100$ a $t_{min} = 0.01$. Počáteční obava z časové náročnosti procesu detekce diskredibility pomocí EA se nepotvrdila. Ve zmíněném jednoduchém experimentu trvalo určení výsledků oběma metodám jen několik minut. I když je zřejmé, že u složitějších zařízeních bude nutné uskutečnit více iterací, a celý proces se stane časově náročnější, diskredibilita senzoru nevede k fatálním chybám v činnosti regulačního obvodu, tento proces je dlouhodobý a časově mnohem náročnější než detekce diskredibility.

Dosažené výsledky opravňují k testování použitelnosti metod pro detekci diskredibility kyslíkové sondy u popsaného kotle na spalování biomasy.


Škála měřených veličin není dostatečná a bude nutné nalézt způsob estimace koncentrace kyslíku ve spalinách. Jako perspektivní se jeví možnost zjišťování pomocí průtoků, tepelného spádu a stechiometrických objemů, což je ovšem předmětem aktuálních výzkumů. Dále je plánována experimentální aplikace na detekci poruch kyslíkové sondy, která je využívána při optimalizaci tohoto procesu spalování. V této realizaci bude úkolem verifikovat, zda sonda poskytuje správné hodnoty. Tento experiment by měl odhalit možnosti pro verifikace kredibility sondy a byl by důležitým krokem k nesimulovaným aplikacím.

Tento výzkum je podporován grantem Ministerstva školství, mládeže a tělovýchovy České republiky MSMT 680770009.


Literatura

- [1] V. Venkatasubramanian, R. Rengaswamy. A Review of Process Fault Detection and Diagnosis. Quantitative Model-based Methods. *Computers & Chemical Engineering*. Vol 27, No 3, pp 293–311, March 2003. ISSN 0098–1354.
- [2] I. Zelinka. *Umělá inteligence v problémech globální optimalizace*. Ben, Praha, 2002, 1. vydání. ISBN 80–7300–06–95.
- [3] M. Nekvida. Genetické algoritmy. In: *Automatizace*, ročník 44, č. 10, s. 614–619, říjen 2001. ISSN 0005–125X.
- [4] D. Klimánek, B. Šulc. Sensor Discredibility Detection via Computational Intelligence. *WSEAS Transactions on Systems*. Vol 4, No 11, pp 1906–1915. October 2005. ISSN 1109–2777.
- [5] V. Mařík, J. Lažanský & O. Štěpánková. *Umělá inteligence 3*. Academia, Praha, 1993. ISBN 80–200–0472–6
- [6] A. Chippefield, P. Fleming. The Matlab Genetic Algorithm Toolbox, in *IEEE Colloquium on Applied control technology using Matlab*, London, England. January, 1995.
- [7] M. Witczak. *Identification and Fault Detection of Non-Linear Dynamic Systems*, University of Zielona Gora Press, Zielona Gora, Poland, 2003. ISBN 80–2162–548–1.
- [8] N. Metropolis, A. W. Rosenbluth & M. N. Rosenbluth. Equation of State Calculation by Fast Computing Machines. *Journal of Chemical Physics*. Vol 21, No 6, pp 1087–1092, May 1999. ISSN 0098–1354
- [9] V. Kvasnička, et al. *Úvod do teórie neuronových sietí*. Iris, Bratislava, 1997. ISBN 80–88778–30–1
- [10] J. Chen, R. Patton. *Robust Model Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publisher, New York, 1999. ISBN 0–7923–8411–3.
- [11] R. King. *Computational Intelligence in Control Engineering*. Basel, New York, 1999. ISBN 0–8247–1999–8.
- [12] S. J. Russell, P. Norvig. *Artificial Intelligence: a Modern Approach*. Prentice–Hall International, Englewood Cliffs, NJ, USA, 1995. ISBN 01–3790–395–2.
- [13] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Cambridge, 1989. ISBN 0–2011–5767–5.
- [14] J. Lažanský, O. Štěpánková & V. Mařík. “Application of the Multi/Agent Approach in Production Planning and Modeling,” in *Proceedings of Engineering Applications of AI*, pp. 369–376, Prague, 2001. ISBN 0–08–043657–9.
- [15] P. Fleming, C. Purshouse. Genetic Algorithms in Control Systems Engineering. In *IFAC Professional Brief – Genetic algorithms in control systems engineering*. 2001, Sheffield, England. [online]
<URL: http://www.oeaw.ac.at/ifac/publications/pbriefs/PB_Fleming_Purshouse.pdf >

- [16] Chen et al. Parallel Genetic Simulated Annealing: A Massively Parallel SIMD Algorithm, in *IEEE Transactions on Parallel and Distributed Systems*. Vol 9, No 2, pp 126-137, February 1998. Los Alamitos, USA. ISSN 1045-9219.
- [17] S. Moins. *Implementation of a Simulated annealing Algorithm for Matlab*. Linköping University Electronic Press, Linköping, Sweden 2002 <URL: <http://www.diva-portal.org/liu/undergraduate/abstract.xsql?dbid=1344>
- [18] European Network of Excellence in Evolutionary Computing. *Evolutionary Strategy Toolbox for Use with MATLAB*. [online] May 2001
<URL: <http://evonet.lri.fr/evoweb/resources/software/record.php?id=319>>
- [19] J. Bašta, et al. *Topenářská příručka: 120 let topenářství v Čechách*. GAS, Praha, 2001. ISBN 80-86176-82-7
- [20] B. Šulc, D. Klimánek. Sensor Discredibility Detection via Computational Intelligence. *WSEAS Transactions on Systems*. Vol 4, No 11, pp 1906-1915. November 2005. ISSN 1109-2777.
- [21] D. Klimánek, B. Šulc. Evolutionary Detection of Sensor Discredibility in Control Loops. *Accepted for the IECON'05, 31st Annual Conference of IEEE on Computer and Control Systems*. November 2005, Raleigh, North Carolina, USA.

Ing. David Klimánek  David.Klimanek@fs.cvut.cz

Ústav řídicí a přístrojové techniky, Fakulta strojní, ČVUT v Praze, Technická 4, Praha, 166 07

Doc. Ing. Bohumil Šulc, CSc.  Bohumil.Sulc@fs.cvut.cz

Ústav řídicí a přístrojové techniky, Fakulta strojní, ČVUT v Praze, Technická 4, Praha, 166 07

Ing. J. Hrdlička.  Jan.Hrdlicka@fs.cvut.cz

Ústav mechaniky tekutin a energetiky, Fakulta strojní, ČVUT v Praze, Technická 4, Praha, 166 07