

OPTIMALIZACE PARAMETRŮ PID REGULÁTORU POMOCÍ GA TOOLBOXU

Radomil Matoušek, Stanislav Lang

Department of Applied Computer Science
Faculty of Mechanical Engineering, Brno University of Technology

Abstrakt

Tento příspěvek se věnuje problematice návrhu optimálního regulátoru s využitím metody optimalizace pomocí tzv. genetického algoritmu (GA). Využití evolučních algoritmů při návrhu parametrů regulátoru je poměrně nové, přičemž možné referenční zdroje je možné hledat například v pracích [Sekaj99, Hypijsova09]. Jako nástroj pro implementaci genetického algoritmu byl zvolen GA toolbox systému Matlab. Bez rozdílu na typ regulátoru (lineární, fuzzy, neuronový) je při jeho návrhu vždy nutné respektovat požadavek na průběh regulačního děje (překmit, rychlost apod.). V tomto příspěvku je na zvoleném modelu demonstrován PID regulátor navržený metodou genetické optimalizace (GA), který je rovněž porovnán s PID regulátorem navrženým pomocí empirické metody Ziegler-Nichols. Jako minimalizační funkce jsou využity klasická integrální kritéria: lineární, kvadratické, ITAE.

1 Obecně k návrhu regulátoru

Návrh regulátoru je bezesporu optimalizační úlohou. Hledanými parametry jsou tvar a koeficienty (operátorového) přenosu regulátoru. I pro oblast řízení a regulace lze uvažovat o možnostech použití evolučních technik pro design regulačních členů. V současné době není uvedený přístup evolučního návrhu parametrů regulátoru příliš rozšířen přesto, že možnosti klasických metod mohou být v některých případech značně omezené. Na druhou stranu je vhodné poznamenat, že pro úlohy, kde si vystačíme s klasickými metodami, nemá smysl hledat jiné řešení, protože klasické metody vedou na exaktní popis (a určí jednoznačné řešení).

Bez ohledu na typ regulátoru (lineární, fuzzy, neuro) je při jeho návrhu klíčovým požadavkem průběh regulačního děje. V některých aplikacích je naprosto nepřijatelný překmit výstupní veličiny, především v oblasti robotiky, kde řízenou veličinou je například poloha robotického ramene. V jiných případech je překmit výstupní veličiny dovolen, ale je požadováno minimální kmitání (tj. požadavek na aperiodický přechodový děj). Dalším hlediskem může být požadavek na rychlost vyregulování výstupní veličiny na žádanou hodnotu. Požadavky na velkou rychlost a minimální překmit jsou ve většině případů protichůdné. Budeme-li se soustředit na metody návrhu klasických PID regulátorů, existuje mnoho přístupů jejich návrhu, například:

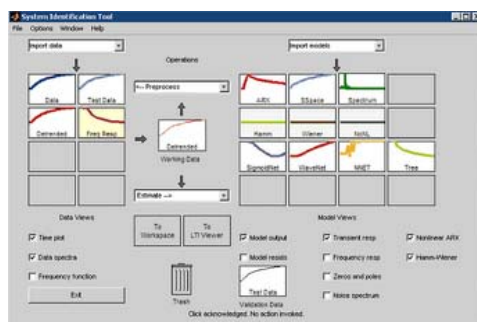
1. Využití frekvenčních charakteristik.
2. Optimální modul.
3. Metoda geometrického místa kořene („Pole placement“).
4. Využití algebraické (polynomiální) teorie řízení.
5. Integrální kritéria (kde e je regulační odchylka.):

a. IAE (lineární)
$$I_{IAE} = \int_0^{\infty} |e(t) - e(\infty)| dt, \quad (1)$$

b. ISE (kvadratické)
$$I_{ISE} = \int_0^{\infty} e^2(t) dt, \quad (2)$$

c. ITAE
$$I_{ITAE} = \int_0^{\infty} |e(t) - e(\infty)| t dt. \quad (3)$$

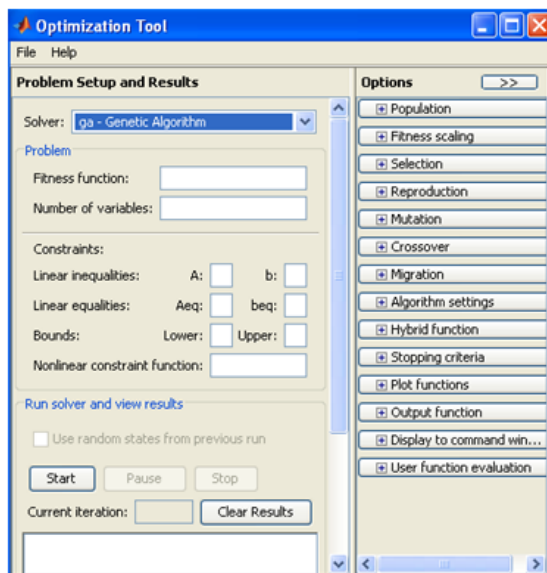
Posloupnost činností vedoucích k návrhu systému řízení lze v podstatě rozdělit do dvou částí. První částí je identifikace systému. Tato oblast je i z hlediska praktické realizace dobře propracovaná a v tomto ohledu můžeme například využít Identification Toolbox systému Matlab.



Obrázek 1: GUI - Matlab Identification Toolbox.

Druhou částí je návrh regulátoru a jeho verifikace. Právě tato část je předmětem tohoto příspěvku, kdy pro návrh parametrů regulátoru PID (PSD) je využito evolučních optimalizačních technik. Verifikace programu je provedena sledováním odezev systému (základní metoda ověření). Verifikace v širším slova smyslu je sama o sobě velmi rozsáhlou oblastí, která přesahuje rozsah tohoto příspěvku, bylo by možné například provádět citlivostní analýzu na změny parametrů systému – v reálném systému může docházet ke kolísání některých parametrů, odolnost řídicího algoritmu vůči takovým změnám se pak označuje jako robustnost.

Užitý Matlab Genetic Toolbox obsahuje implementaci evolučních optimalizačních algoritmů jako je genetický algoritmus, simulované žíhání aj. Využití toolboxu nabízí odladěné algoritmy a odbourává tak nutnost jejich programování, čímž vytváří prostor pro intenzivnější soustředění se na jejich aplikaci v dané oblasti. Genetický algoritmus patří k nejrozšířenějším evolučním algoritmům. Užitý Genetic toolbox nabízí uživateli poměrně široké možnosti nastavení parametrů samotného genetického algoritmu, počet jedinců v populaci, způsob výběru jedince pro křížení, typ a pravděpodobnost mutace a další.



Obrázek 2: GUI - Optimalizační toolbox rozšířený o evoluční algoritmy.

2 Optimalizace regulátoru pomocí genetického algoritmu

V příspěvku prezentujeme výsledky evolučního návrhu regulátoru pro systém 2. řádu s dopravním zpožděním. Genetický algoritmus je implementován v nadstavbě prostředí MatLab Genetic toolbox jako funkce, jejímiž vstupními parametry jsou mimo jiné *ukazatel na optimalizovanou funkci* a *počet optimalizovaných parametrů*. Prvním krokem je proto vytvoření funkce reprezentující daný optimalizační problém. Dále je potřeba nastavit volitelné vlastnosti genetického algoritmu (počet

jedinců v populaci, typ výběru jedince pro křížení, typ křížení, typ a četnost mutací, atd.). V neposlední řadě je pak zapotřebí reprezentovat výsledky formou přijatelnou pro uživatele. Realizaci lze tedy rozdělit na dva základní kroky:

1. Vytvořit účelovou (hodnotící) funkci tak, aby její vstupní parametry měly význam parametrů regulátoru a její výstupní hodnota „fitness“ vypovídala o kvalitě regulace. Hodnotící funkce je realizována jako funkce prostředí MatLab v souboru typu m-file, a byla pojmenována *simulacee*. Volání funkce má tvar:

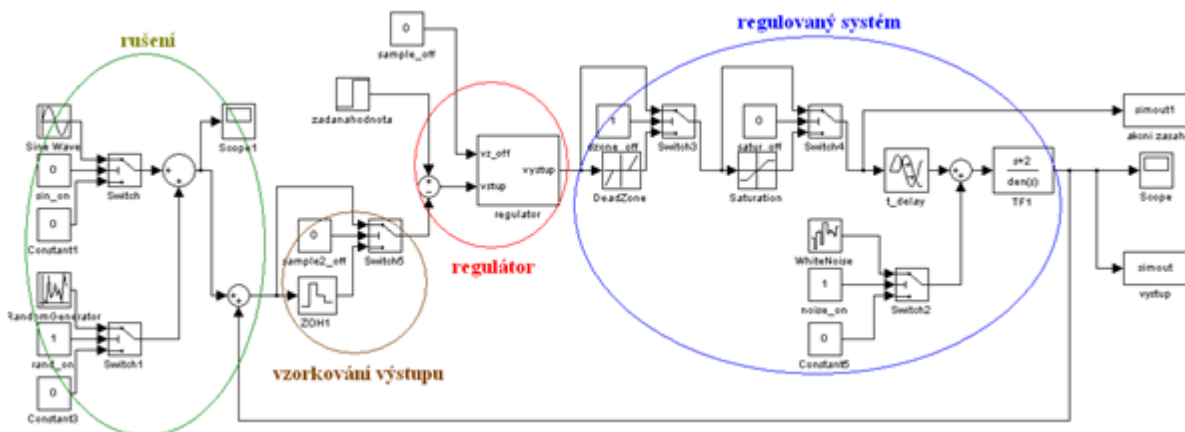
$$fitness_jedince = simulacee(parametr1,parametr2,...)$$

Z funkce je volána simulace v prostředí Matlab/Simulink. Simulační model představuje zpětnovazební regulační smyčku obsahující zadaný systém a regulátor, jehož parametry nejsou konstanty ale proměnné, takové se kterými byla funkce *simulacee* volána.

2. Optimalizace dle navržené účelové funkce:
 - a. nastavení fixních parametrů účelové funkce,
 - b. nastavení parametrů optimalizačního algoritmu,
 - c. reprezentace výsledků.

3 Simulační model

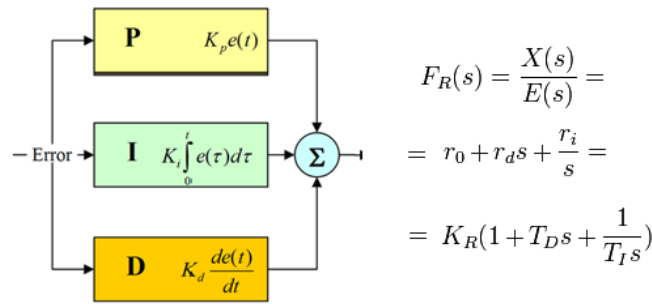
Základem simulačního modelu je zpětnovazební smyčka obsahující regulátor a regulovaný systém. Regulovaný systém se skládá z lineární části přenosu *TF1*, dopravního zpoždění *t_delay*, saturace *Saturation*, pásma necitlivosti *DeadZone* a šumu *WhiteNoise* vstupujícího do systému. Parametry všech zmíněných bloků (detail níže) jsou nastaveny staticky před začátkem optimalizace. Lineární část systému je tvořena přenosovou funkcí, koeficienty polynomu čitatele *num(s)* a jmenovatele *den(s)* jsou zadány jako dva vektory reálných čísel reprezentujících koeficienty polynomů. Časové zpoždění je zadáno jako nezáporné reálné číslo (zpoždění může být i nulové, tj. žádné). U saturace definujeme horní a dolní mez, u pásma necitlivosti začátek a konec pásma, u šumu jeho výkon, všechny tři uvedené bloky lze přemostit (pro připojení resp. přemostění slouží prepínače *Switch*).



Obrázek 3: Simulační model

Regulátor PID (ve své spojité i diskrétní obdobe) je velmi rozšířený a charakteristický svou značnou robustností, mimo jiné obsahuje integrační složku, čímž zajistí nulovou regulační odchylku. Spojitý PID regulátor najdeme mezi nástroji simulačního prostředí Simulink. Pro popis bloku využívá Simulink první z forem složkového tvaru.

$$F_R(s) = K + I / s + D \cdot s \tag{4}$$



Obrázek 4: Vnitřní zapojení PID regulátoru a jeho operátorový přenos.

Na základě výsledků simulace je spočítána hodnota *fitness*, tedy vhodnost jedince reprezentujícího konkrétní kombinaci nastavení parametrů regulátorů. Ještě před spuštěním samotné simulace je ověřeno, zda je kombinace parametrů přípustná (např. není-li délka vzorkovací periody záporná, apod.), pokud jsou parametry nekorektní, simulace spuštěna není a vrací se rovnou velká hodnota *fitness* (tak, aby výrazně převyšovala *fitness* ostatních jedinců). Pokud jsou parametry korektní a simulace proběhla, průběh výstupu a akčního zásahu byly uloženy do pracovních proměnných (typu pole). Základní hodnotu *fitness* spočítáme podle jednoho (vybraného) integrálního kritéria regulace. Navíc aby byla postižena míra vysokofrekvenčního kolísání akčního zásahu, je k hodnotě *fitness* přičtena suma absolutních hodnot druhé diference a hodnoty obdobně spočítané. Pomocí proměnné „váha“ lze nastavit míru potlačení kmitání (použita hodnota „1“). Markantnější kolísání penalizujeme mnohem výraznějším způsobem, hodnota *fitness* je vynásobena počtem průchodů střední hodnotou (ideální stav je jeden průchod).

$$fitness = fitness + \frac{\left(\sum_k |\Delta_2(k)| + \sum_k |\Delta_{2m}(k)| \right)}{k} \cdot vaha$$

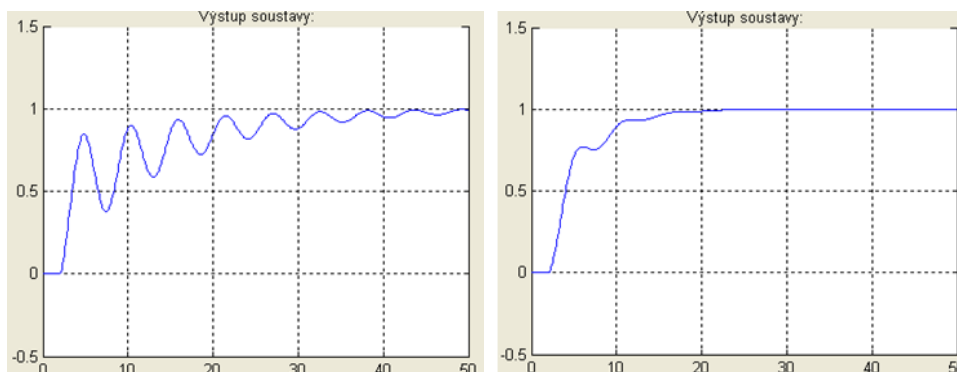
$$\Delta_2(k) = x(k-1) - 2x(k) + x(k+1) \quad (5)$$

$$\Delta_{2m}(k) = x(k-1) + x(k) - 2x(k+1)$$

4 Testování a výsledky optimalizace

Po úspěšné kontrole parametrů, je spuštěn proces optimalizace. Počáteční populace je vygenerována náhodně, nejen proto je průběh nalezených řešení vždy jiný (počáteční podmínky jsou odlišné, během prohledávání působí mutace atd.), většinou však dříve či později algoritmus dospěje k podobným výsledkům. Rychlost konvergence je ovlivněna (nejen) velikostí populace a nastavením parametrů genetického algoritmu. Regulační děj je typicky v prvních krocích velmi kmitavý a postupně je „usměrňován“. Uvedme ilustrační příklad regulace soustavy s přenosem:

$$F(s) = \frac{s+2}{s^3 + 6s^2 + 6s + 5} \cdot e^{-3s} \quad (6)$$



Obrázek 5: Nejlepší nalezené řešení po 2. iteraci (vlevo), resp. po 48. iteraci (vpravo).

5 Závěr

Prezentovaný příspěvek naznačuje cestu možného nastavení parametrů regulátoru reálné soustavy s využitím evolučních algoritmů, resp. genetického algoritmu implementovaného prostřednictvím GA toolboxu systému Matlab. Na základě provedených experimentů je možné konstatovat, že genetický algoritmus je použitelnou optimalizační metodou pro nalezení parametrů regulátoru. S příslušným GUI se může jednat o jeden z flexibilních nástrojů vhodných k nastavení parametrů regulátoru na základě zvolených parametrů.

The screenshot shows a GUI for configuring a control system. It is divided into three main sections:

- Soustava (System):** Contains options for defining the system transfer function. The first option is "polynomů B a A" (polynomials B and A), which is selected. Below it, there are input fields for the coefficients of polynomials B and A. The coefficients for B are [1 2 3 4 0.5] and for A are [1 6 6 5]. These fields are circled in red. Other options include "zesílení, nul a pólů" (gain, zeros, and poles) with fields for gain (1), zeros (empty), and poles (-1).
- Další parametry soustavy (Further system parameters):** Contains various checkboxes and input fields for system characteristics: "Dopravní zpoždění" (transport delay) set to 3 s; "Saturace" (saturation) checked, with min -15 and max 15; "Pásmo necitlivosti" (deadband frequency) with start -0.01 and end 0.01; "Bílý šum vstupující do soustavy" (white noise entering the system) set to 0.0005; "Náhodná chyba měřená výstupu" (random measurement error) set to 0.005; "Vzorkování regulátoru" (regulator sampling) with "zadat: Ts=" (specify: Ts) set to 0.1 s; "Vzorkování snímače na výstupu" (sensor sampling at output) checked, with "Ts2=" set to 0.1 s.
- Požadavky (Requirements):** Contains optimization criteria: "Kritérium pro optimalizaci" (optimization criterion) with "Kritérium ITAE" selected; "Dovolit překmit" (allow overshoot) unchecked; "Maximální dovolený překmit" (maximum allowed overshoot) set to 0.01.

A yellow banner at the bottom of the GUI displays the message "systém není kauzální" (system is not causal). At the very bottom, there are options to "zobrazit:" (display) "akční zásah" (control action) or "fitness nejlepšího" (fitness of the best).

Obrázek 6: Příklad uživatelského interface.

Poděkování

Práce vznikla jako součást řešení standardního SPP FSI BD13001025.

References

- [1] Lang, S.: Diplomová práce - Optimalizace řídicího algoritmu pomocí evolučního algoritmu. VUT v Brně, Brno, ČR, 2010.
- [2] Švarc, Ivan; Šeda, Miloš; Vítečková, Miluše. *Automatické řízení*. 1. vyd. Brno : AKADEMICKÉ NAKLADATELSTVÍ CERM, s.r.o., 2007. 324 s. ISBN 978-80-214-3491-2.
- [3] Jura, Pavel. *Signály a systémy : Část 3 Diskrétní signály a diskrétní systémy*. Brno : VUT, FEKT, ÚAMT, 2003. 89 s.
- [4] Pivoňka, Petr. *Optimalizace regulátorů*. Brno : VUT, FEKT, ÚAMT, 2005. 112 s.

Adresa:

ÚAI, Fakulta strojního inženýrství, VUT v Brně, Technická 2, 616 69, Brno, ČR
email: {matousek, lang}@fme.vutbr.cz