Настройка на размити ПИД регулатори чрез генетичен алгоритъм в MATLAB и SIMULINK

Донка Иванова

Genetic-based tuning for fuzzy PID controllers in MATLAN and SIMULINK: *This paper describes a new method for the design of fuzzy PID controllers based on genetic optimization. The proposed controller is with very simple structure. It uses a one input fuzzy inference with three rules and six tuning parameters. The proposed control system has been applied to several first-order with time delay, second order, third-order with time delay and fifth-order processes. Simulation results show that proposed fuzzy PID controller produces closely or superior control performance than the conventional PID controllers, tuned also by genetic algorithms.*

Key words: *Fuzzy logic control, PID control, Genetic algorithms, Optimal control*

ВЪВЕДЕНИЕ

Генетичният алгоритъм (ГА) е стохастичен метод за решаване на оптимизационни задачи с и без ограничения, основан на естествената селекция, процесът определящ биологичната еволюция. ГА променя на всяка стъпка популацията от индивиди, като избира тези от текущото поколение, които ще продължат развитието си, т.е. ще се използват за следващата генерация. Селекцията на най-добрите индивиди става въз основа на функционал или функционали (целеви функции), даващи оценка за близостта на индивида с желаното решение. Правилата по които става изборът на индивиди от текущата популация са:

- правила за селекция – избират се най-добрите индивиди, които да участват в създаването на следващото поколение или да бъдат прехвърлени без промяна в следващото поколение;
- правила за кръстосване – избраните чрез селекция индивиди се кръстосват, като се цели да се получат индивиди, които да наследят най-добрите характеристики на родителите си;
- правила за мутация – чрез случайна промяна на някои от гените се гарантира, дори и нито един от индивидите в текущото поколение да не съдържа необходимия ген, пак да се достигне екстремум.

Генетичният алгоритъм може да се приложи за решаване на различни оптимизационни задачи, за решаването на които не са подходящи стандартните оптимизационни методи, включително и такива, при които целевата функция е прекъсната, недиференцируема, стохастична или силно нелинейна.

В много разработки се разглеждат хибридни системи, които комбинират различни интелигентни методи, а именно генетичен алгоритъм с размита логика [5,6,7,8].

Размитата логика намира все по-голямо приложение при управление на различни технологични процеси. Това определя необходимостта от изучаване на размитите регулатори и съпоставянето им с регулаторите, реализиращи типови непрекъснати закони за регулиране. Съществуват многобройни методи за избор и настройка на линейни непрекъснати регулатори [1, 2]. Проектирането на размити регулатори, в повечето случаи, се извършва чрез компютърна симулация. Много разработки се отнасят до реализацията на размито ПД, ПИ и ПИД управление, избор на структура и сравнение на размитите с класическите регулатори [4, 9]. Обособени са две групи параметри на размитите регулатори - структурни и параметри за настройка [5]. Към структурните параметри, определяни off-line, се отнасят входно-изходните променливи и функциите им на принадлежност, базата от размити правила, механизмът за получаване на размити изводи и механизмът за деразмиване. Към параметрите за настройка, определяни on-line, се отнасят коефициентът на пропорционалност, коефициентите на интегралната и диференциалната съставки, параметрите на функциите на принадлежност и мащабните коефициенти. Следователно проектирането на размити ПИД регулатори може да се разглежда като оптимизационна задача, решението на която дава оптималните стойности на параметрите за настройка на регулаторите.

Целта на настоящата статия е да се приложи генетичният алгоритъм за настройка на размити ПИД регулатори и се сравнят процесите с тези, получени в системи с линейни непрекъснати ПИД регулатори, настроени чрез генетичен алгоритъм.

ПРОЕКТИРАНЕ НА РАЗМИТИ ПИД РЕГУЛАТОРИ

Целта при проектиране на размити ПИД регулатори е да се постигне желаното управление чрез регулатор с възможно най-проста структура и най-малко на брой параметри за настройка. На фиг.1 е представена структурата на размит ПИД регулатор с една входна и една изходна променливи [5].

Входна променлива за размития регулатор е грешката $e(n) = r(n) - y(n)$ ($r(n)$ е задаващото въздействие, $y(n)$ е изходната променлива на системата), а изходна променлива е $u(n)$. Нормализираното управляващо въздействие е

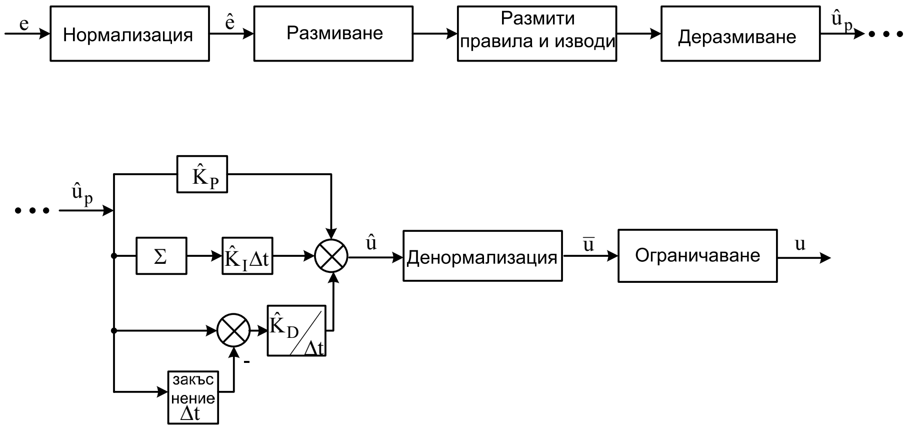
$$\hat{u}(n) = \hat{K}_P \hat{u}_P(n) + \hat{K}_I \sum_{i=0}^{n} \hat{u}_P(i)\Delta t + \hat{K}_D \frac{\Delta \hat{u}_P(n)}{\Delta t}, \quad n = 0,1,2,..., \tag{1}$$

където $\hat{K}_P, \hat{K}_I$ и $\hat{K}_D$ са нормализираните коефициенти за пропорционалната, интегралната и диференциалната съставки.

Нормализираната грешка се определя като

$$\hat{e}(n) = \begin{cases} 1, & s_e e(n) > 1 \\ s_e e(n), & |s_e e(n)| \le 1 \\ -1, & s_e e(n) < -1, \end{cases} \tag{2}$$

където $s_e = |1/(r(0) - y(0))|$, $r(0)$ и $y(0)$ са началните стойности на задаващото въздействие и на управляемата величина.



Фиг.1. Структура на размит ПИД регулатор

Денормализацията на управляващото въздействие $\hat{u}$ е по

$$\bar{u}(n) = s_u \hat{u}(n), \tag{3}$$

където s_u е коефициентът на денормализация.

За управляващото въздействие се записва

$$u(n) = \begin{cases} u_{max}(n), & \bar{u}(n) > u_{max} \\ \hat{u}(n), & u_{min} \le \bar{u}(n) \le u_{max} \\ u_{min}(n), & \bar{u}(n) < u_{min}, \end{cases} \tag{4}$$

където u_{min} и u_{max} са възможните минимална и максимална стойности на входната променлива на обекта за управление.

Разглежданият размит регулатор е с три правила

П1: If ($\hat{e}$ is NB) then ($\hat{u}_P$ is NB)

П2: If ($\hat{e}$ is PB) then ($\hat{u}_P$ is PB)

П3: If ($\hat{e}$ is AZ) then ($\hat{u}_P$ is AZ),

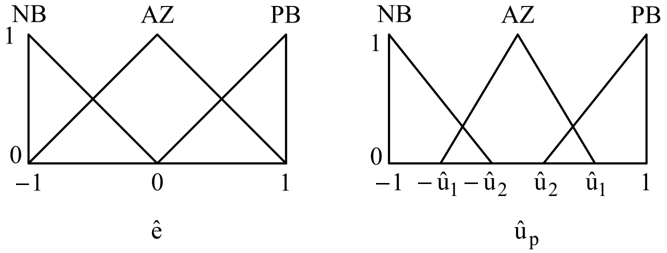
като размитата променлива “NB” е за „отрицателно големи”, “PB” – за „положително големи” и “AZ” – за „приблизително нула” стойности. Функциите на принадлежност за променливите $\hat{e}$ и $\hat{u}_P$ са избрани триъгълни, като за $\hat{e}$ са фиксирани параметрите, а за $\hat{u}_P$ се търсят оптималните стойности, фиг. 2.

Следователно разглежданият размит регулатор има само три правила и шест параметъра за настройка като

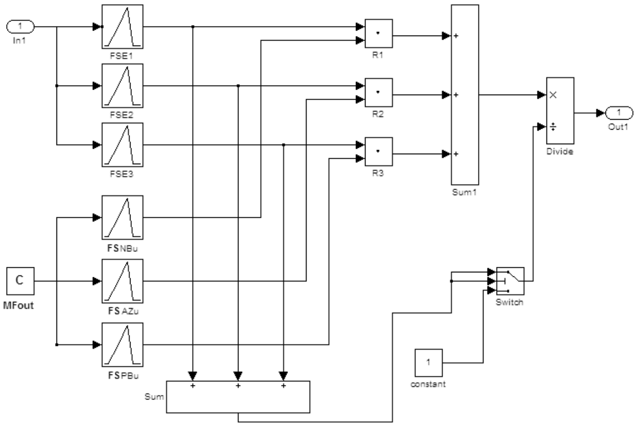
$$\begin{aligned} &0 < \hat{K}_P \le 1, \quad 0 < \hat{K}_I \le 1, \quad 0 < \hat{K}_D \le 1, \\ &0 < s_u \le max(|u_{min}|, |u_{max}|), \\ &0 < \hat{u}_1 \le 1, \quad 0 < \hat{u}_2 \le 1. \end{aligned} \tag{5}$$

За реализиране на размития регулатор не може да се използва FIS редактора на Fuzzy Logic Toolbox, защото там параметрите на функциите на принадлежност се въвеждат с конкретните си числени стойности. Затова размитият регулатор е моделиран в MATLAB\SIMULINK, фиг.3 [3]. На входа In1 се подава сигналът $\hat{e}$, а на изхода Out1 се получава $\hat{u}_p$.

Параметрите на триъгълните функции на принадлежност FSE1, FSE2 и FSE3 на входната променлива са [-2, -1, 0], [-1, 0, 1] и [0, 1, 2] (лява, връх и дясна стойност на триъгълната функция на принадлежност) за NB, AZ и PB съответно. Параметрите на функциите на принадлежност на изходната променлива са [-2u_2, -1, -u_2], [-u_1, 0, u_1] и



Фиг.2. Функции на принадлежност на променливите $\hat{e}$ и $\hat{u}_P$



Фиг.3. SIMULINK модел на размития регулатор

[u_2, 1, 2u_2] за NB, AZ и PB съответно, като конкретните стойности на u_1 и u_2 се определят чрез оптимизационната процедура. Трите правила на размития регулатор са моделирани чрез блоковете R1, R2 и R3. След разделяне на сумата от изходите на блоковете R1, R2 и R3 със сумата от изходите на функциите на принадлежност се извършва деразмиване по метода на центъра на тежеста.

НАСТРОЙКА НА РАЗМИТИ ПИД РЕГУЛАТОРИ ЧРЕЗ ГЕНЕТИЧЕН ЛГОРИТЪМ

Настройката на размития ПИД регулатор се свежда до решаване на оптимизационна задача при зададен критерий J_T. Използването на критерии, основани само на сигнала на грешката в системата трудно биха удовлетворили противоречивите изисквания между точността в установен режим и динамичните показатели на системата – времетраене и пререгулиране. Затова все по-често се използват не един, а няколко критерия J_i, отчитащи особеностите и изискванията на съответната задача. Решаването на получената задача за векторна оптимизация може да стане по различни начини. Най-често се прилага тегловният метод, при който задачата се свежда до оптимизация при един критерий $J_T = \sum_i \omega_i J_i$.

Тегловните коефициенти ω_i се използват, ако има голяма разлика между стойностите на отделните критерии J_i, или ако желаем някой да има по-голяма тежест. Връзката между обобщения критерий J_T и параметрите за настройка на размития ПИД регулатор е нелинейна, което определя целесъобразността от прилагане на генетичния алгоритъм за настройката му.

В настоящата работа е използван критерият [5]

$$J_T = \omega_1 \frac{\int_0^T e^2(t)dt}{max(e(n))} + \omega_2.\sigma + \omega_3 \frac{t_p}{T}, \tag{6}$$

където t_p е времетраенето на преходния процес, s;

σ - пререгулирането, %;

T - времето за симулация, s.

При единично стъпаловидно входно въздействие се предполага, че $max(e(n))$=1.

Оптимизационната задача за настройка на размития ПИД регулатор е решена в MATLAB и SIMULINK. В SIMULINK е направен модел на затворената система за автоматично регулиране. Въведени са променливите $\hat{u}_1$, $\hat{u}_2$ за функциите на принадлежност на изходната променлива на размития регулатор и променливите $\hat{K}_P, \hat{K}_I$ и $\hat{K}_D$, определящи нормализираните пропорционална, интегрална и диференциална съставки. В блок Gain е въведена променливата s. Функцията, изчисляваща критерия за оптимизация е записана в M-file (evaluation_JT). В него са обявени глобалните променливи $\hat{K}_P, \hat{K}_I, \hat{K}_D, \hat{u}_1, \hat{u}_2, s$, чрез командата sim се извиква SIMULINK модела и се изчислява целевата функция. Генетичният алгоритъм се стартира чрез M-file или в команден режим на MATLAB, чрез командите [10]:

```
>> global K_P K_I K_D û_1 û_2 s
>> options=gaoptimset;
>> options=gaoptimset(options,'MutationFcn',@mutationadaptfeasible);
>> options=gaoptimset(options,'PlotFcns',@gaplotbestf);
>> options=gaoptimset(options, 'StallTimeLimit',250);  % Възможна е промяна и на други
                                         настройки на оптимизационната процедура
>> LB=[0 0 0 0 0 0.001];
>> UB=[1 1 1 1 1 10];
>> [X,fval]=ga(@evaluation_JT,6,[],[],[],[],LB,UB,[],options)
```

Векторите LB и UB определят долната и горната граници на параметрите за оптимизация $\hat{K}_P, \hat{K}_I, \hat{K}_D, \hat{u}_1, \hat{u}_2, s$.

Оптимизационната процедура се прекратява при достигане на зададения толеранс при изчисляване на целевата функция (options.TolFun).

ПРИМЕРИ ЗА НАСТРОЙКА НА РАЗМИТ ПИД РЕГУЛАТОР ЧРЕЗ ГЕНЕТИЧЕН АЛГОРИТЪМ

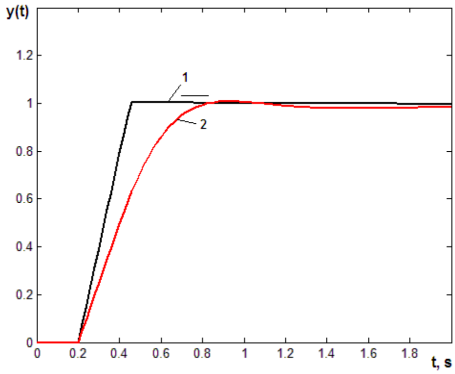
Моделирани са системи за автоматично управление на различни обекти с размити ПИД и ПИД регулатори, настроени чрез генетичен алгоритъм. Предавателните функции на обектите за управление са:

- Пример 1 - $W(p)=\dfrac{1}{p+1}e^{-0,2p}$, $u_{min}=0$, $u_{max}=10$;
- Пример 2 - $W(p)=\dfrac{2}{p^2+4p+3}$, $u_{min}=0$, $u_{max}=5$;
- Пример 3 - $W(p)=\dfrac{1}{(p+1)^3}.e^{-5p}$, $u_{min}=0$, $u_{max}=5$;
- Пример 4 - $W(p)=\dfrac{1}{(p+1)^5}$, $u_{min}=0$, $u_{max}=5$.

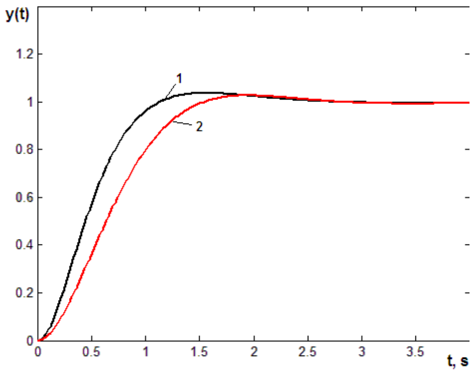
Стойностите на параметрите за настройка на ПИД и размития ПИД регулатори, стойността на целевата функция J_T, пререгулирането σ и времетраенето t_p на преходните процеси в моделираните системи са приведени в Таблица 1. Преходните процеси в системите с размит ПИД регулатор, крива 1, и ПИД регулатор, крива 2, са показани на фиг. 4, фиг. 5, фиг. 6 и фиг. 7. Очевидно е, че чрез размития ПИД регулатор се получават близки или по-добри показатели на

Таблица 1

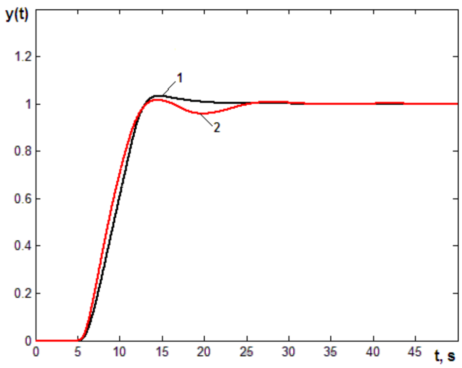
Обект на управление	Регу-латор	$\hat{K}_P$	$\hat{K}_I$	$\hat{K}_D$	s	$\hat{u}_1$	$\hat{u}_2$	J_T	σ	t_p
Пример 1	Размит ПИД	1	0,916	0	4,501	0,951	0,939	10,555	0	0,44
	ПИД	1	0,874	0	2,52	-	-	12,617	0,82	0,7
Пример 2	Размит ПИД	1	0,8	0	7,56	0,811	0,082	9,522	3,68	0,97
	ПИД	1	0,891	0	2,49	-	-	10,083	2,7	1,32
Пример 3	Размит ПИД	0,190	0,141	0.109	1,08	0.42	0,960	19,495	3,29	12,38
	ПИД	0,53	0,110	0,879	1	-	-	19,983	1,62	12,25
Пример 4	Размит ПИД	0,28	0,205	0,158	2,42	0,42	0,960	16,124	7,35	10,6
	ПИД	0,631	0,116	0,866	2,227	-	-	17,666	3,2	11,8



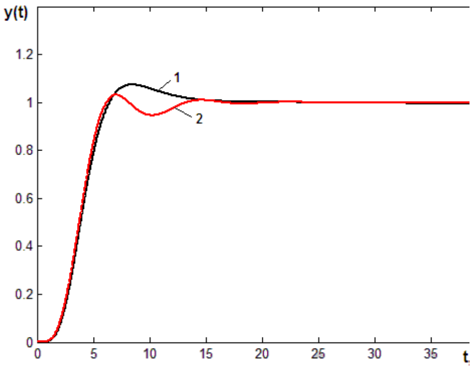
Фиг.4. Преходни процеси в системата от Пример 1



Фиг. 5. Преходни процеси в системата от Пример 2



Фиг. 6. Преходни процеси в системата от Пример 3



Фиг. 7. Преходни процеси в системата от Пример 4

качеството на преходните процеси в сравнение с тези, получени в системите с ПИД регулатор, настроен чрез генетичен алгоритъм.

ЗАКЛЮЧЕНИЕ

Използването на генетичния алгоритъм за настройка на размити ПИД регулатори дава много добри резултати за системи без и със закъснение. Разгледаният размит регулатор е с много проста структура – една входна и една изходна променливи, само три размити правила и шест параметъра за настройка.

Анализът на преходните процеси в системите с размити ПИД регулатори показва, че показателите на качеството са близки или по-добри в сравнение с тези в системите с ПИД регулатори, настроени чрез генетичен алгоритъм. Разгледаните размити регулатори ще осигурят по-добра робастност на системите, отколкото съответните линейни непрекъснати регулатори.

ЛИТЕРАТУРА

1. Гарипов, Е. Проектиране на стандартни аналогови ПИД регулатори в Matlab и SIMULINK. // Автоматика и информатика, № 1, 2004, с. 35 – 42.
2. Костов, К., Е. Николов. Инженерни емпирични методи за настройка – II. // Автоматика и информатика, № 4, 2002, с. 52 – 54.
3. Atlas, A. I. H., A. M. Sharaf. A Generalized Direct Approach for Designing Fuzzy Logic Controllers in Matlab/Simulink GUI Environment. International Journal of Information Technology and Intelligent Computing, Int. J.It &IC, №4, vol.1, 2007.
4. Chen, C.–L., F.–C Kuo. Design and Analysis of a Fuzzy Logic Controllers. International Journal of Systems Science, vol. 26, 1995, pp. 1223 – 1248.
5. Hu, Baogang, G. K. I. Mann, R. G. Gosine. New Methodology for Analytical and Optimal Design of Fuzzy PID Controllers. IEE Transactions on Fuzzy Systems, vol.7, № 5, 1999, pp. 521 – 539.
6. Kim, J., Y. Moon, B. P. Zeigler. Designing fuzzy net controllers using genetic algorithms, IEE Cont. Syst. Mag., 1995, pp. 66-72.
7. Linkens, D. A., H. O. Nyongesa. Genetic algorithms for fuzzy control, part 1: Offline system development and application, Proc. Inst. Elect. Eng-Control Theory Appl., vol. 142, 1995, pp. 161-176.
8. Linkens, D. A., H. O. Nyongesa. Genetic algorithms for fuzzy control, part 1: Online system development and application, Proc. Inst. Elect. Eng-Control Theory Appl., vol. 142, 1995, pp. 177-185.
9. Petrov, M., A. Taneva, I. Ganchev. Fuzzy PID Controller Design in Matlab Environment. Proceedings of the Int. Conf. Automatics and Informatics, Sofia, 2007, pp. 57 – 61.
10. www.mathworks.com/access/helpdesk/help/.../gads/

За контакти:

Доц. д-р Донка Иванова, Катедра “Автоматика, информационна и управляваща техника”, Русенски университет “Ангел Кънчев”, тел.: 082-888 226, e-mail: divanova@uni-ruse.bg

Докладът е рецензиран.