Marian B. GORZAŁCZANY, Adam GŁUSZEK
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, KIELCE UNIVERSITY OF TECHNOLOGY
Al. 1000-lecia Państwa Polskiego 7, 25-314 Kielce

Designing fuzzy rule-based controllers from data using particle swarm optimization

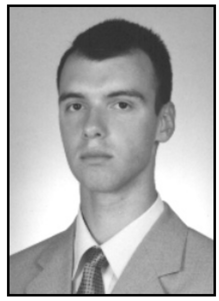
Prof. Marian B. GORZAŁCZANY

In 1978 he received the M.Sc. degree in electronics from Warsaw Univ. of Technology, in 1983 and 1994 – the Ph.D. and D.Sc. degrees in computer science from Poznań Univ. of Technology and in 2003 – the professor title. In 1990-1992 he was working at Univ. of Saskatchewan and Univ. of Guelph, Canada. He is the author/co-author of more than 80 refereed scientific works including the monograph published by Springer-Verlag. The main area of scientific interests is designing and application of computational intelligence systems.

e-mail: m.b.gorzalczany@tu.kielce.pl

Ph.D. eng. Adam GŁUSZEK

He was born in 1975. He graduated from Department of Electrical and Computer Engineering, Kielce University of Technology (specialization: electrical engineering) in 1999. He received his Ph.D. degree in computer science from Systems Research Institute of Polish Academy of Sciences in 2001. The area of his scientific interests covers computational intelligence, including neural networks, fuzzy systems, evolutionary algorithms and multi-agent systems.



e-mail: a.gluszek@tu.kielce.pl

Abstract

The paper presents a hybrid combination of fuzzy rule-based systems and particle swarm optimization (PSO) approach – referred to as PSO fuzzy rule-based technique – for optimizing sets of fuzzy control rules synthesized from control data. The application of the proposed technique to a complex and non-linear problem of the control of backing up a truck to a loading dock is also presented in the paper.

Keywords: computational intelligence, fuzzy controllers, multi-agent systems, swarm intelligence, measurement data.

Projektowanie rozmytych regulatorów regułowych na bazie danych z wykorzystaniem tzw. optymalizacji rojowej

Streszczenie

Artykuł prezentuje hybrydowe połączenie rozmytych systemów regułowych z metodami tzw. optymalizacji rojowej w celu optymalizowania zestawów rozmytych reguł sterujących syntetyzowanych z danych opisujących procesy sterowania. Zatem, artykuł proponuje rozszerzenie tradycyjnego zestawu komponentów wykorzystywanych dotychczas w budowie systemów tzw. inteligencji obliczeniowej obejmującego sztuczne sieci neuronowe, systemy rozmyte, algorytmy ewolucyjne (przede wszystkim, algorytmy genetyczne) czy też tzw. zbiory przybliżone o nowe narzędzie przeszukiwania rozważanych przestrzeni rozwiązań. Najpierw sformułowany został problem budowy regulatorów, których funkcjonowanie opisywane jest przy pomocy zestawów reguł rozmytych, na bazie danych opisujących procesy sterowania. Następnie przedstawiono zarys procesu syntezy rozmytych reguł sterowania z wykorzystaniem metod tzw. optymalizacji rojowej. Z kolei, zaprezentowano zastosowanie proponowanego podejścia do złożonego i nieliniowego problemu sterowania cofaniem ciężarówki do rampy załadowczej. Przedstawiono uzyskaną bazę reguł regulatora rozmytego, kształty funkcji przynależności zbiorów rozmytych występujących w regułach sterowania oraz wybrane przykłady trajektorii ruchu ciężarówki ze sterowaniem rozmytym.

Słowa kluczowe: inteligencja obliczeniowa, regulatory rozmyte, systemy wieloagentowe, inteligencja rojowa, dane pomiarowe.

1. Introduction

As already mentioned in paper [5] included in this volume, data (also measurement data) can be a very useful and comprehensive source of information that describe the operation of many complex systems including also control systems. Due to vast amounts of data available nowadays (see e.g. numerous databases, Internet resources, etc.), intensive research efforts that aim at creating "intelligent" tools for designing models of complex systems from data are underway. It is associated with intelligent data analysis and discovery of the knowledge (in the form of various patterns, trends, decision mechanisms, etc.) in the data. A distinct research area that addresses the above-mentioned issues are the so-called computational intelligence (henceforward: CI) systems, cf. [2], [8]. They are based on hybrid and synergistic combinations of such techniques as artificial neural networks, fuzzy logic, evolutionary computations, rough sets, etc.

In this paper the authors propose to extend the typical set of CI components by the so-called particle swarm optimization (henceforward: PSO) technique (see e.g. [1, 6, 7]) from the area of multi-agent systems. In general, the performance of PSO is based on moving simple software agents (the so-called particles) within the search space. Each particle, which is associated with a candidate solution, searches for a better position in the search space by changing its velocity according to the rules inspired by natural behaviour of animal flocks.

The paper presents a hybrid combination of fuzzy rule-based systems and PSO approach – referred to as PSO fuzzy rule-based technique – for optimizing sets of control rules synthesized from control data. First, the problem of the fuzzy rule-based controller design using data is formulated. Then, a PSO-based learning of fuzzy rules from data is outlined. Finally, the application of the proposed technique to a complex and non-linear problem of the control of backing up a truck to a loading dock is presented.

2. Designing a fuzzy rule-based controller from data

Consider a fuzzy controller with n inputs $x_1, x_2, ..., x_n$ ($x_i \in X_i$, $i = 1,2,...,n$) and one output y ($y \in Y$). Assume that the behaviour of the controller is described by K input-output data records:

$$L = \{\boldsymbol{x}'_k, y'_k\}_{k=1}^{K}, \tag{1}$$

where $\boldsymbol{x}'_k = (x'_{1k}, x'_{2k}, ..., x'_{nk}) \in \boldsymbol{X} = X_1 \times X_2 \times ... \times X_n$, $y'_k \in Y$, k is the number of input-output data record and K is the total number of such records. Additionally, let $L_{\boldsymbol{x}} = \{\boldsymbol{x}'_k\}_{k=1}^{K} \subset \boldsymbol{X}$.

Designing the rule-based controller from data consists in:

1. Finding a mapping $M: \boldsymbol{X} \to Y$ provided its restriction on data L (1) (referred to as the learning data): $M_L: L_{\boldsymbol{x}} \to Y$ is known.
2. Formulating and tuning a set of fuzzy rules that model – in a readable and easily-interpretable way – the operation of the considered controller. These rules implement the mapping M of point 1.
3. Pruning the obtained fuzzy rule base, that is, removing superfluous, "weaker" rules (this improves the transparency and interpretability of the controller) and analyzing how it affects the accuracy of its functioning (this addresses the problem of a trade-off between the controller's performance and interpretability).

The fuzzy control rules that will be synthesized from data (1) in point 1 and then analyzed in point 3 have the following form:

$$\text{IF } (x_1 \text{ is } A_{1r}) \text{ AND} \ldots \text{AND } (x_n \text{ is } A_{nr}) \text{ THEN } (y \text{ is } B_r), \quad (2)$$

where $A_{ir} \in F(X_i)$, $i = 1,2,...,n$, and $B_r \in F(Y)$ are the S-, M-, or L-type fuzzy sets (see [4] for details) representing verbal terms Small, Medium and Large, respectively, in the r-th fuzzy rule, $r = 1,2,...,R$. $F(X_i)$ and $F(Y)$ denote families of all fuzzy sets defined in the universes X_i and Y, respectively. The inputs and the output of the controller can be described by one S-type, one L-type and several M-type fuzzy sets (see Section 4 of this paper for details on the shapes of their membership functions).

The fuzzy rules (2) are being processed according to most widely used fuzzy approach, that is, the aggregation of the rule antecedents is performed by a *t*-norm of the *minimum*-type, the same *t*-norm is used to represent the IF-THEN rules, a *t*-conorm of the *maximum*-type is used to aggregate the fuzzy rules, and the compositional rule of inference is employed to generate the controller's decisions, cf. [2].

3. PSO-based learning of fuzzy rules from data

Before the learning begins, an initial fuzzy rule base of the form (2) is created from the learning data (1) using the widely used "look-up table" algorithm [9]. Then, during the PSO-based learning, the parameters of the input and output S-, M-, and L-type fuzzy sets of rules (2) are tuned in order to minimize the assumed objective function.

Following [1, 6, 7], an outline of the PSO approach will now be presented. In PSO, the so-called swarm is composed of a set of particles $S = \{s_1, s_2, \ldots, s_J\}$. At any time step t, a particle s_j, $j = 1,2,...,J$ has a position $\boldsymbol{p}_j^t$ and a velocity $\boldsymbol{v}_j^t$ associated to it. The position $\boldsymbol{p}_j^t$ of the particle s_j represents a candidate solution of the considered optimization problem with an objective function f. The best position $\boldsymbol{p}_j^t$ (that is, with the minimal value of $f(\boldsymbol{p}_j^t)$) that particle s_j has ever visited is represented by $\boldsymbol{b}_j^t$ and referred to as the particle's *personal best*. Moreover, the best position ever found by any particle in the swarm is represented by $\boldsymbol{g}_j^t$ (it is also called the *global best*). The PSO algorithm starts by generating random positions for all the particles within the considered search space. Velocities are initialized to small random values. During the main loop of the algorithm, the velocities and positions of the particles are iteratively updated until an assumed stopping criterion is met. The update formulas are:

$$\boldsymbol{v}_j^{t+1} = \omega \boldsymbol{v}_j^t + \varphi_1 \boldsymbol{U}_1^t (\boldsymbol{b}_j^t - \boldsymbol{p}_j^t) + \varphi_2 \boldsymbol{U}_2^t (\boldsymbol{g}_j^t - \boldsymbol{p}_j^t), \quad (3)$$

$$\boldsymbol{p}_j^{t+1} = \boldsymbol{p}_j^t + \boldsymbol{v}_j^{t+1}, \quad (4)$$

where ω is a parameter called *inertia weight* (or *momentum*), φ_1 and φ_2 are parameters called *acceleration coefficients*, and $\boldsymbol{U}_1^t$ and $\boldsymbol{U}_2^t$ are diagonal matrices in which the entries in the main diagonals are random numbers uniformly distributed in the interval $[0,1)$. These matrices are regenerated at each iteration. The comments regarding the interpretation of coefficients ω, φ_1 and φ_2 can be found in the afore-mentioned references. The common termination condition in the PSO algorithm is reaching the assumed number of iterations or finding a solution with acceptable value of the objective function.

4. Fuzzy rule-based control of backing up a truck

Backing up a truck to a loading dock is a non-linear control problem, which is difficult to solve with the use of conventional methods. The proposed PSO fuzzy rule-based approach will be used to design a fuzzy controller for backing up a simulated truck in a planar parking lot. Simulation experiments have been performed in the same convention as in [8]. The controller has two input variables (x_1 – the x-position coordinate x, $-150 \le x \le 150$, and $x_2 = \phi$ – the angle of the truck with the vertical, $-180 \le \phi \le 180$) and one output ($y = \theta$ – the steering angle, $-45 \le \theta \le 45$). At every stage of the control, the controller produces the steering angle that backs up the truck to the loading zone from any initial position and from any angle in the loading zone.

The fuzzy controller has been designed using the learning data:

$$L = \{x'_{1k}, x'_{2k}, y'_k\}_{k=1}^{K=282}, \quad (5)$$

which is a collection of correct truck trajectories, that is, the samples of truck positions ($x'_1 = x$ and $x'_2 = \phi$) and steering signals $y' = \theta$, provided by an expert. The expert has the knowledge coming from experience regarding the truck behaviour and is able to successfully control the truck without using any particular formal model of the control system.

Before the proper learning begins, an initial fuzzy rule base of the form (2) with $n = 2$ is created from the learning data (5) using the afore-mentioned "look-up table" algorithm [9]. For each input and for the output, five fuzzy sets representing verbal terms Small, Medium1, Medium2, Medium3 and Large have been defined. In the learning phase, the PSO algorithm with the swarm of 100 particles, acceleration coefficients φ_1 and φ_2 equal to 1.2 and maximal number of iterations equal to 500 has been used. The inertia weight ω has been linearly decreased on the learning horizon from the initial value equal to 0.4 to the final – equal to 0. During the learning process, the parameters (central points and widths) of the membership functions of input and output fuzzy sets have been tuned to minimize the assumed objective function. This function is an aggregated mean squared error between desired and actual activation degrees of the controller's output fuzzy sets – see [2] for details.

Fig. 1 presents the plot of the objective function for the best and average particles (individuals) versus the number of iterations.

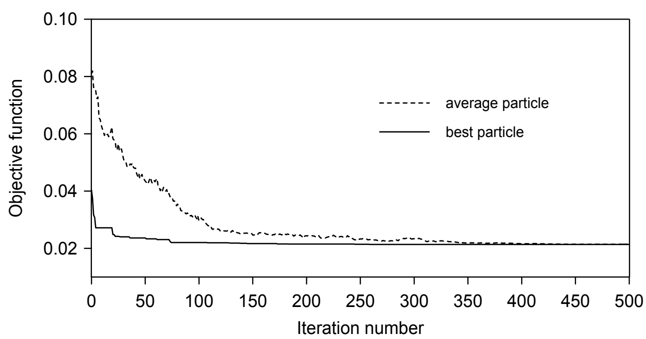


Fig. 1. Objective function vs. iteration number plot
Rys. 1. Przebieg wartości funkcji celu w trakcie procesu uczenia

Fig. 2 shows the final shapes of the membership functions of fuzzy sets describing the controller's inputs: x_1 – Fig. 2a, x_2 – Fig. 2b and output y – Fig. 2c. Table 1 presents the full rule base generated by the proposed approach.

The obtained fuzzy controller performs successfully for any initial position of the simulated truck – Fig. 3 shows selected examples of simulated truck trajectories in the closed-loop control system.

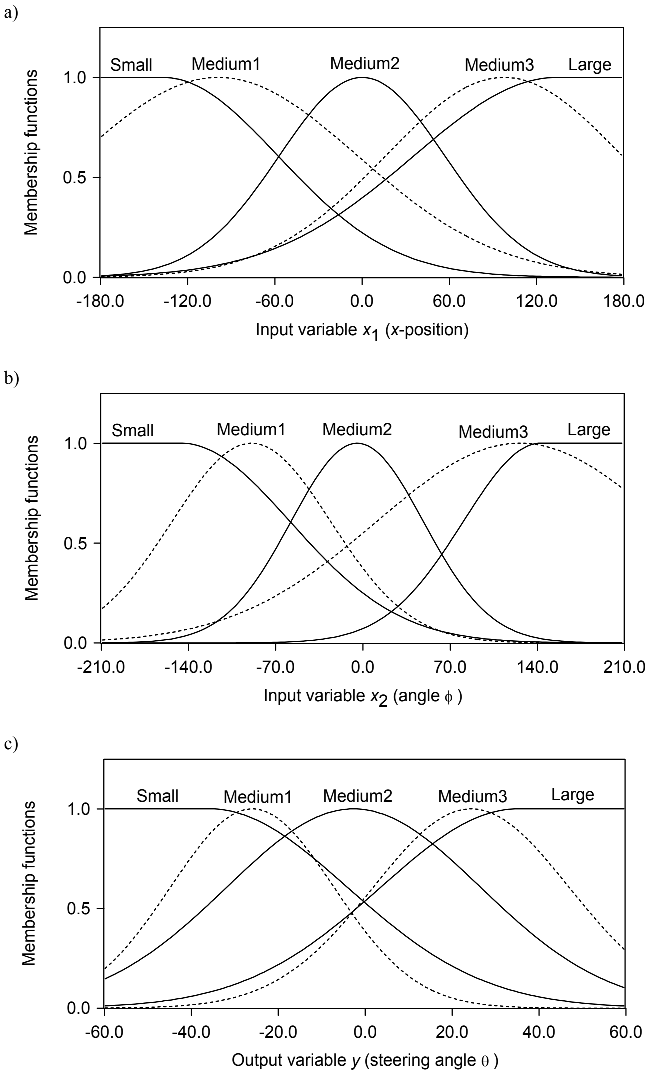


Fig. 2. Final shapes of membership functions of fuzzy sets describing controller's inputs: x_1 (a), x_2 (b) and output y (c)
Rys. 2. Końcowe kształty funkcji przynależności zbiorów rozmytych opisujących zmienne wejściowe: x_1 (a), x_2 (b) i zmienną wyjściową y (c) regulatora

Tab. 1. Final fuzzy rule base of the controller
Tab. 1. Końcowa baza reguł rozmytych regulatora

$x_2=\phi$ / $x_1=x$	S	M1	M2	M3	L
S	L	L			M2
M1	L	L	S	M3	L
M2		S	M2	L	
M3	S	M1	L	S	S
L	M2			S	S

S=Small, M1=Medium1, M2=Medium2, M3=Medium3, L=Large $y=\theta$

5. Conclusions

The combination of fuzzy rule-based systems and particle swarm optimization approach for optimizing sets of control rules synthesized from control data has been briefly presented in this paper. The performance of the proposed technique has also been successfully verified in a complex and non-linear problem of the control of backing up a truck to a loading dock. This problem is difficult to solve with the use of conventional methods.

The approach presented in this paper as well as the genetic fuzzy modelling and evolution-strategy fuzzy modelling presented in papers [5] and [3] (in this volume), respectively, demonstrate that the computational intelligence techniques provide powerful tools for intelligent data analysis in general and, in particular, for data-based modelling and control of complex systems and processes.

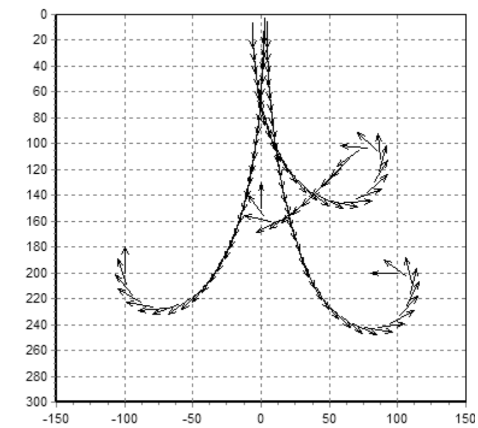


Fig. 3. Selected examples of truck trajectories under fuzzy control
Rys. 3. Wybrane przykłady trajektorii ciężarówki ze sterowaniem rozmytym

6. References

[1] Engelbrecht A.P.: Fundamentals of Computational Swarm Intelligence. John Wiley and Sons, Chichester, UK, 2005.
[2] Gorzałczany M.B.: Computational Intelligence Systems and Applications, Neuro-Fuzzy and Fuzzy Neural Synergisms. Physica-Verlag, Springer-Verlag Co., Heidelberg, New York, 2002.
[3] Gorzałczany M.B., Głuszek A.: Data-based fuzzy modelling of dynamic systems by means of evolution strategies. Pomiary Automatyka Kontrola, vol. 56, no. 12, 2010, pp. 1427 - 1429.
[4] Gorzałczany M.B., Rudziński F.: A modified Pittsburg approach to design a genetic fuzzy rule-based classifier from data, in L. Rutkowski, R. Scherer, R. Tadeusiewicz, L.A. Zadeh, J. Żurada (Eds.), Artificial Intelligence and Soft Computing – ICAISC 2010, Part I, Lecture Notes in Artificial Intelligence 6113, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 88-96.
[5] Gorzałczany M.B., Rudziński F.: Measurement data in genetic fuzzy modelling of dynamic systems. Pomiary Automatyka Kontrola, vol. 56, no. 12, 2010, pp. 1420 - 1423.
[6] Kennedy J, Eberhart R.: Swarm Intelligence. Morgan Kaufman, San Francisco, CA, 2001.
[7] Kennedy J, Eberhart R.: Particle Swarm Optimization. Proc. of IEEE International Conference on Neural Networks, IEEE Press, Piscataway, NJ, 1995.
[8] Kosko B.: Neural Networks and Fuzzy Systems, Prentice-Hall, Englewood Cliffs, 1992.
[9] Wang L.-X., Mendel J.M.: Generating fuzzy rules by learning from examples, IEEE Trans. System, Man, Cybernetics, vol. 22, no. 6, pp. 1414-1427, 1992.

otrzymano / received: 25.09.2010
przyjęto do druku / accepted: 01.11.2010
artykuł recenzowany