Fuzzy controller synthesis algorithm with complete data on the object state

Svetlana Luneva[1,*] and *Anna* Luneva[1]

[1]Moscow Aviation Institute (National Research University), 4, Volokolamskoe shosse, Moscow, 125993, Russia

Abstract. The application of the classical control theory to nonlinear and linear objects control problems in some cases may not give effective results. Due to this, fuzzy controller synthesis methods for objects controlling have been developed based on fuzzy sets and fuzzy logic methods. A suggested method makes it possible to construct a fuzzy system without the help from human experts. The algorithm for this method consists of three steps: on step 1 the known input data ranges are determined and divided into fuzzy regions to form membership functions; on step 2 the data pairs are formed and then used to generate a fuzzy rules table; on step 3 the membership functions are scaled for specific values, so the fuzzy system works more correct. The method has shown it's good performance in the underwater vehicle stabilization problem with complete data on the object state.

1 Introductions

Fuzzy systems synthesis method consists in fuzzification of input and output data, formation of fuzzy rules table by human experts in the field of the problem being solved and its subsequent use, fuzzy data processing with the resulting fuzzy system and its defuzzification [1, 2, 7, 8, 11, 12].

This paper is devoted to the solving of fuzzy controller synthesis problem with complete data on the object state in the stabilization of an underwater vehicle problem. The solution is based on the algorithm of the generating rules from the known numerical data method, which makes it possible to construct a set of fuzzy rules [3].

2 Statement of the problem

An underwater vehicle moving in a vertical plane at a constant longitudinal travel speed [4, 5] is considered as a control object (see figure 1).

The mathematical model of the control object is described by a system of differential equations

$$\begin{cases} \dot{\omega} = a_{11}\omega + a_{12}\psi + b\delta, \\ \dot{\psi} = \omega, \\ \dot{\delta} = u \end{cases}$$

with the given initial conditions: $\omega(0) = \omega_0$, $\psi(0) = \psi_0$, $\delta(0) = \delta_0$.

[*] e-mail: LunevaSY@mai.ru

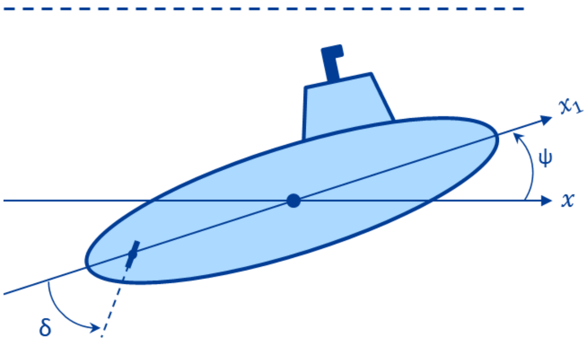


Figure 1. Control object

Here ψ—trim angle, δ—aft horizontal rudders deviation angle from the neutral position, ω—angular velocity, u—control.

The position of the underwater vehicle is characterized by a state vector consisting of three coordinates $x(t) = (\omega(t), \psi(t), \delta(t))^T$. The control is carried out through one channel.

The base initial values are considered as $\omega(0) = 0$, $\psi(0) = 4$, $\delta(0) = 0$ [4, 5].

The task here is to find the control $u(t, x)$ and its corresponding trajectories of motion $x(t)$, that ensure the object bringing to a horizontal position for any initial conditions from a given set of initial states.

One of the possible solutions to this problem is to minimize the quadratic criterion $I = \int_0^\infty [B_1 \cdot \omega(t)^2 + B_2 \cdot \psi(t)^2 + B_3 \cdot u(t)^2]\, dt$ using an optimal linear controller $u(t, x) = = K_1\omega(t) + K_2\psi(t) + K_3\delta(t)$. The procedure of finding the controller is reduced to finding a positive definite solution to the algebraic Riccati equation [4]. As a result, the coefficients of the controller K_1, K_2, K_3 are found.

Let's consider the problem solution using fuzzy logic: we have to synthesize a fuzzy controller with three input variables and one output variable ψ, δ, ω using the method of generating fuzzy rules based on known data.

3 The algorithm of generating fuzzy rules from known data method

Step 1. Setting fuzzy variables and their fuzzy values

For each variable, we define the domain intervals in which its values will probably lie during the control system functioning. Each domain interval is then divided into N generally intersecting regions for input variables ω, ψ, δ and into M regions for the output variable u, and each region is assigned with a triangular or trapezoidal fuzzy membership function (figures 2(a)–(d)). The membership function determines the corresponding linguistic meaning [10]: P1 (Psy 1), ..., PN (Psy N), D1 (Delta 1), ..., DN (Delta N), O1 (Omega 1), ..., ON (Omega N) for the input variables C1 (Control 1), ..., CM (Control M) for the output variable.

We assign linguistic variables $\tilde{\psi}$, $\tilde{\delta}$, $\tilde{\omega}$ to each input variable and a linguistic variable $\tilde{u}$ to the output variable. Their linguistic meanings are defined as: $\tilde{A}_i = \{\tilde{A}_i^j, j = 1, \dots, 2N + 1\}$, $\tilde{B} = \{\tilde{B}^j, j = 1, \dots, 2M + 1\}$.

Step 2. Generating fuzzy rules from given data pairs

To apply the generating fuzzy rules method, the known data must be represented as pairs, where the first element of the pair is the state vector, and the second is the value of control [3, 6, 9]. The values corresponding to the vertices of the membership functions and determining the linguistic values of the coordinates are selected as possible values for each coordinate of the state vector. To form fuzzy rules, all possible combinations of these values are fed to the input of the fuzzy system, and the control value is calculated for each of these combinations. Then the values of the corresponding membership functions are found, and the largest of them is selected. The found highest value corresponds to a certain linguistic meaning of control. If the values of two neighboring membership functions are equal, if the corresponding membership functions lie to the left of zero, the value corresponding to the right membership function and the left one in the opposite case is selected. Thus, one linguistic rule is assigned to each data pair.

2.1. Forming data pairs

2.1.1. To generate data pairs, the values of the state vector variables corresponding to the vertices of the membership functions are taken and all possible combinations are composed.

2.1.2. The values of the control u corresponding to each combination are calculated. Thus, data pairs, where the first element of the pair is the state vector, and the second element of the pair is the control value u are formed.

2.1.3. The obtained values of the variables are written into separate ordered sets, where the ordinal numbers of the elements of the set correspond to the ordinal number of the data pairs obtained on step 2.1.2.

2.2. Computing the values of membership functions for each data pair
The calculation of the values of the membership functions is iterative. At each iteration:

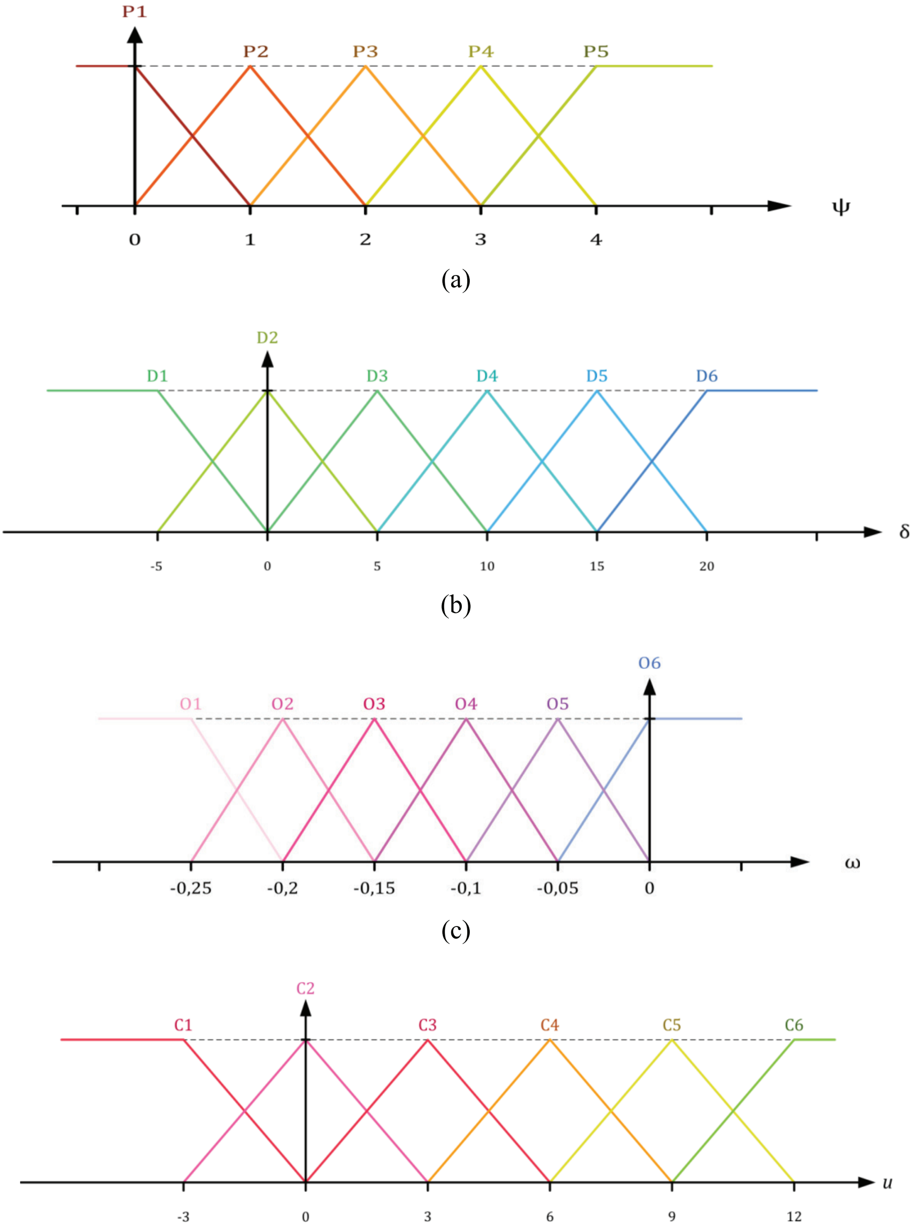
2.2.1. The values of each membership function are calculated for each value of the state vector and control from the corresponding sets created on step 2.1, and then written into separate ordered sets. The values of the membership functions are calculated sequentially, in accordance with their ordinal numbers j. Writing to a set occurs in the same order.

2.2.2. The maximum value among the values of the control membership functions from the obtained set is calculated. Then the ordinal number of the membership function with the maximum value is written into the set to generate the rules. After that, the set with already calculated the membership functions values is cleared to write new values for the next iteration.

2.3. Forming the rule tables according to the principle: IF $\tilde{\psi} = \tilde{A}_1^j$ AND $\tilde{\delta} = \tilde{A}_2^j$ AND $\tilde{\omega} = \tilde{A}_3^j$, THEN $\tilde{u} = \tilde{B}^i$.

The j numbers of linguistic values correspond to the ordinal numbers of the membership functions. When compiling the rules, the sets obtained on step 2.2 are used, where the ordinal numbers of the elements of each set coincide with the number of the rule.

Since there are usually lots of data pairs, and each data pair generates one rule, it is highly possible that there will be some conflicting rules, i.e., rules that have the same “if” part but a different “then” parts [3]. One way to resolve this conflict is to assign a degree to each of the conflicting rules. In practice the additional information about the data pair utility is known, i.e., it is known which data pairs are very useful and generate advice rules and which data pairs are unlikely to be used for generating rules. In this case we can assign a weight coefficient to every conflicting data pair, representing our confidence in generated



(a)

(b)

(c)

(d)

Figure 2. Membership functions for ψ ($N_1 = 5$) (a), δ ($N_2 = 6$) (b), ω ($N_3 = 6$) (c), u ($M = 6$) (d)

rule correctness. In this way, we'll get a fuzzy set, where every data pair belongs to this set with a chosen weight coefficient. Thus in case in case of conflict a rule that has a maximum degree is accepted, and the other ones are deleted.

Since there are three input variables, the rule table can be presented as a parallelepiped. For ease of use, it is convenient to present it in the form of flat sections sequence, i.e. the obtained linguistic rules are ordered in the form of a set of two-dimensional tables with fixed

$\tilde{\omega}$ = «O1»							
		$\tilde{\delta}$					
		D1	**D2**	**D3**	**D4**	**D5**	**D6**
$\tilde{\psi}$	**P1**	C1	C1	C1	C1	C1	C1
	P2	C1	C1	C1	C1	C1	C1
	P3	C2	C2	C1	C1	C1	C1
	P4	C3	C3	C2	C1	C1	C1
	P5	C4	C3	C3	C2	C2	C1

$\tilde{\omega}$ = «O2»							
		$\tilde{\delta}$					
		D1	**D2**	**D3**	**D4**	**D5**	**D6**
$\tilde{\psi}$	**P1**	C1	C1	C1	C1	C1	C1
	P2	C2	C1	C1	C1	C1	C1
	P3	C3	C2	C2	C1	C1	C1
	P4	C4	C3	C2	C2	C1	C1
	P5	C4	C4	C3	C3	C2	C2

$\tilde{\omega}$ = «O3»							
		$\tilde{\delta}$					
		D1	**D2**	**D3**	**D4**	**D5**	**D6**
$\tilde{\psi}$	**P1**	C1	C1	C1	C1	C1	C1
	P2	C2	C2	C1	C1	C1	C1
	P3	C3	C3	C2	C1	C1	C1
	P4	C4	C3	C3	C2	C2	C1
	P5	C5	C4	C4	C3	C3	C2

$\tilde{\omega}$ = «O4»							
		$\tilde{\delta}$					
		D1	**D2**	**D3**	**D4**	**D5**	**D6**
$\tilde{\psi}$	**P1**	C2	C1	C1	C1	C1	C1
	P2	C3	C2	C1	C1	C1	C1
	P3	C4	C3	C2	C2	C1	C1
	P4	C4	C4	C3	C3	C2	C2
	P5	C5	C5	C4	C4	C3	C3

$\tilde{\omega}$ = «O5»							
		$\tilde{\delta}$					
		D1	**D2**	**D3**	**D4**	**D5**	**D6**
$\tilde{\psi}$	**P1**	C2	C2	C1	C1	C1	C1
	P2	C3	C2	C2	C1	C1	C1
	P3	C4	C3	C3	C2	C2	C1
	P4	C5	C4	C4	C3	C3	C2
	P5	C6	C5	C5	C4	C4	C3

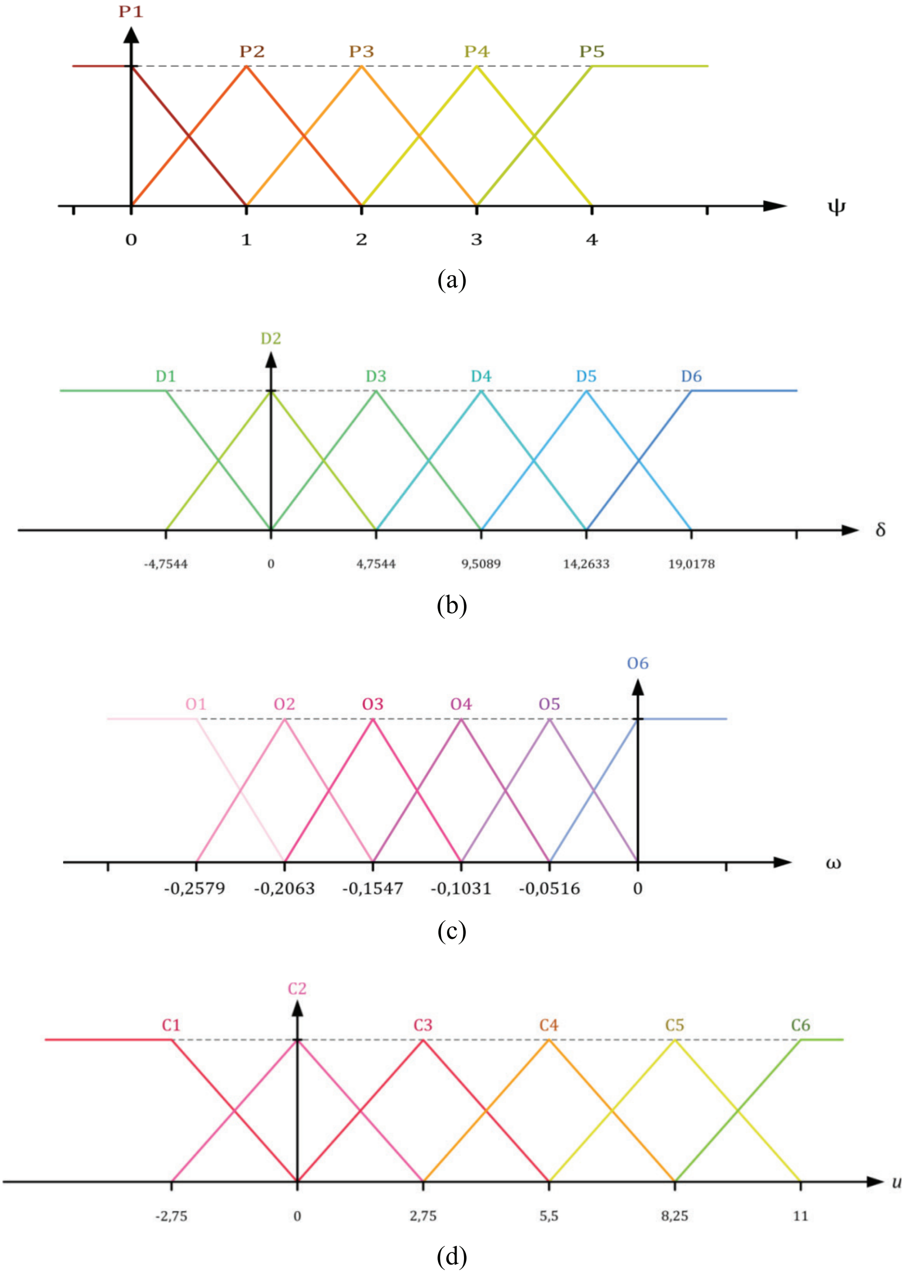
$\tilde{\omega}$ = «O6»							
		$\tilde{\delta}$					
		D1	**D2**	**D3**	**D4**	**D5**	**D6**
$\tilde{\psi}$	**P1**	C3	C2	C1	C1	C1	C1
	P2	C3	C3	C2	C2	C1	C1
	P3	C4	C4	C3	C3	C2	C2
	P4	C5	C5	C4	C4	C3	C2
	P5	C6	C6	C5	C5	C4	C3

Figure 3. Rule table for $\tilde{\omega}$

linguistic values of one of the three input linguistic variables. The obtained rule table for is presented on figure 3.

Step 3. Scaling membership functions

After the table of linguistic rules is compiled, it is saved and remains unchanged for any combination of initial conditions belonging to the specified neighborhoods around the base values. However, the parameters of the membership functions constructed in step 1 of the algorithm must be scaled for specific values taken by the input and output variables during the operation of the system (figures 4(a)–(d)).



(a)

(b)

(c)

(d)

Figure 4. Scaled membership functions for ψ ($N_1 = 5$) (a), δ ($N_2 = 6$) (b), ω ($N_3 = 6$) (c), u ($M = 6$) (d)

The problem was solved numerically using an optimal linear controller and using a synthesized fuzzy controller (figure 5). The fuzzy inference system was implemented using FuzzyLogicToolbox in the MATLAB computer mathematics system, the “min” rule was used to find the intersection of the sets, the “max” rule was used for combining, the “min” rule was used for the implication, and the “centroid” method was used for defuzzification.

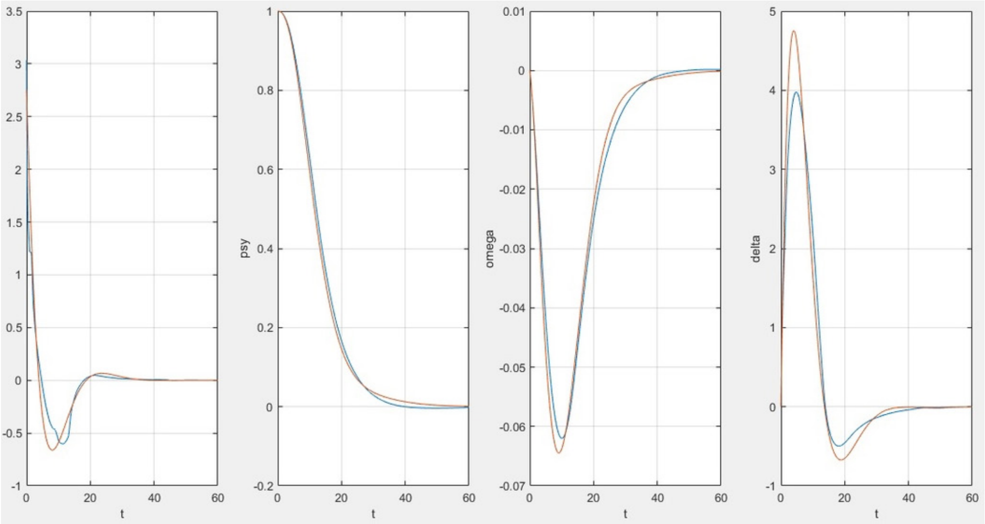


Figure 5. Simulation results for $\omega(0) = 0$, $\psi(0) = 1$, $\delta(0) = 0$

The following parameters were used in the simulation: $a_{11} = -0.1253$; $a_{12} = -0.004637$; $b = -0.002195$; $B_1 = 1.97$; $B_2 = 0.03$; $B_3 = 0.05$; $K_1 = 26.1$; $K_2 = 2.75$; $K_3 = -0.399$.

4 Conclusion

The simulation results comparison showed that the proposed algorithm for the synthesis of a fuzzy controller gives acceptable results and can be used to solve similar problems in conditions of uncertainty of the plant parameters.

References

[1] K.M. Passino, S. Yurkovich, *Fuzzy control* (Addison Wesley Longman, Menlo Park, CA, 1998)

[2] A.G. Badalova, A.V. Panteleev, *Enterprise Risk Management* (Vuzovskaya kniga, Moscow, 2017)

[3] L.X. Wang, J.M. Mendel, *Generating Fuzzy Rules by Learning from Examples*, IEEE Transactions on Systems, Man, and Cybernetics, **22(6)**, 1414 (1992)

[4] E.I. Veremej, *Line systems with feedback* (Lan', Saint-Petersburg, 2013)

[5] A.V. Panteleev, A.S. Bortakovskij, *Control Theory in Examples and Tasks* (Vysshaya shkola, Moscow, 2003)

[6] D. Meng, Z. Pei, *Extracting Linguistic Rules from Data Sets Using Fuzzy Logic and Genetic Algorithms*, Neurocomputing, **78**, 48 (2012)

[7] E.H. Mamdani, S. Assilian, *An experiment in linguistic synthesis with a fuzzy logic controller*, International Journal of Man-Machine Studies, **7(1)**, 1 (1975)

[8] J.M. Alonso, C. Castiello, C. Mencar, *Interpretability of fuzzy systems: Current research trends and prospects*, in Springer Handbook of Computational Intelligence (Springer, Berlin, Heidelberg, 2015)

[9] S. Guillaume, *Designing fuzzy inference systems from data: an interpretability-oriented review*, IEEE Trans. Fuzzy Syst., **9(3)**, 426 (2001)

[10] U. Bodenhofer, P. Bauer, *A formal model of interpretability of linguistic variables*, in Interpretability Issues in Fuzzy Modeling (Springer, 2003)

[11] W. Siler. J.J. Buckley, *Fuzzy expert systems and fuzzy reasoning* (John Wiley & Sons, Hoboken, New Jersey, 2005)

[12] G. Feng, *A survey on analysis and design of model-based fuzzy control systems*, IEEE Trans. Fuzzy Syst., **14(5)**, 676 (2006)