

Chapter 2

Introduction

Preface

As part of the Fuzzy Inference System [?], the Matlab environment offers some reliable tools, which allow characterising fuzzy problems. These include:

- The FIS (Fuzzy Inference System) Editor,
- The Membership Function Editor,
- The Rule Editor,
- The Rule Viewer,
- The Surface Viewer,
- The ANFIS (Adaptive Neuro-Fuzzy Inference System).

The three first editors are used to precisely define a fuzzy problem in the Mamdani or Takagi-Sugeno structure. The System's fourth and fifth element constitute tools allowing visualisation of the decisive area and System operation. The sixth editor makes it possible to build a fuzzy model only within the Takagi-Sugeno structure with parameters selected by a neural network.

2.1 FIS Editor

The FIS (Fuzzy Inference System Editor), which belongs to basic tools, makes it possible to do the following:

1. selecting the fuzzy model category (Mamdani or Takagi-Sugeno),
2. specifying the number of input variables and attributing names to them,

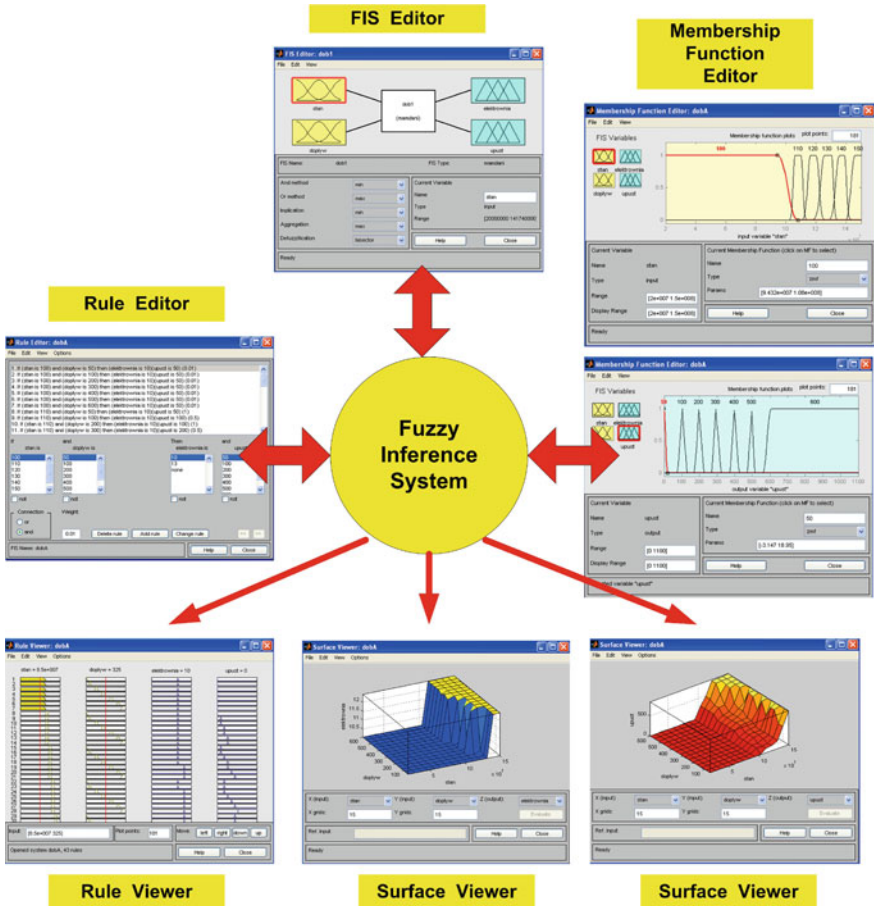


Fig. 2.1 Fuzzy System structure

3. specifying the number of output variables and attributing names to them,
4. selecting the type of T-norm, S-norm, method of implication, aggregation and defuzzification (in the case of the Mamdani model).

Generally, the number of input variables $X(t)^T = [x_1(t), x_2(t), \dots, x_m(t)]$, $t \in [0, T]$ and output variables $Y(t)^T = [y_1(t), y_2(t), \dots, y_n(t)]$ are not limited by the FIS editor; however, the rule base size and the number of analysed operations required for the $X(t)$, $Y(t)$ vector defuzzification rapidly grow with increasing measures of vectors $Y(t)$. Depending on the computer's RAM memory resources and CPU parameters, the speed of processing intended to obtain sharp values of the $Y(t)$ output variables may differ greatly (Figs. 2.1, 2.2 and 2.3).

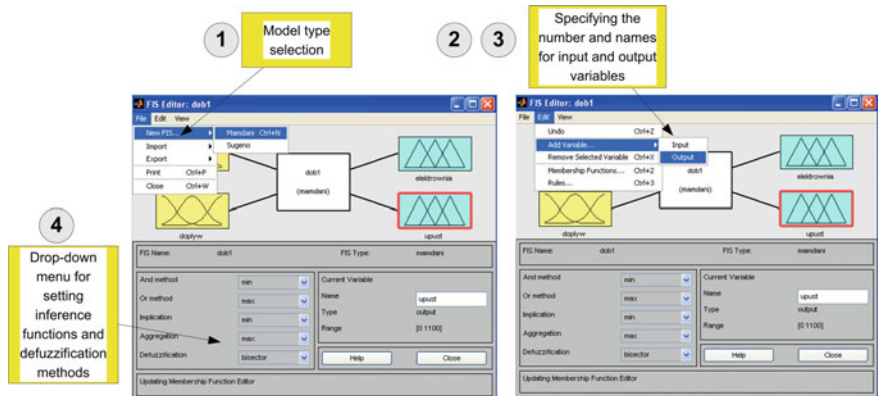


Fig. 2.2 The FIS (Fuzzy Inference System) editor

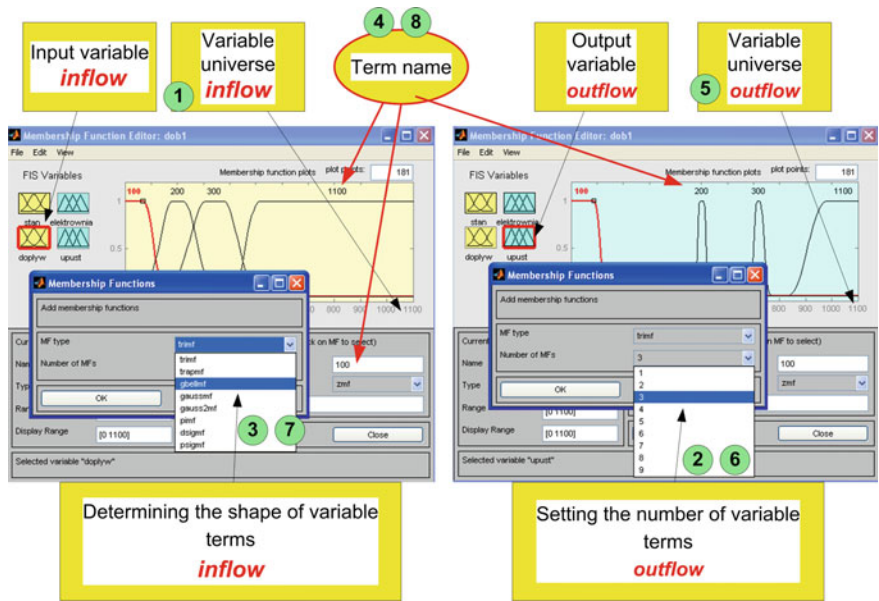


Fig. 2.3 Editor of input and output functions *Membership Function Editor*

When the Matlab application is started, the FIS will be loaded to the operating area after entering the fuzzy instruction, which activates the editor and displays it on the screen ready to formulate the fuzzy problem.

2.2 Membership Function Editor

By double-clicking an input or output variable icon, we activate the Membership Function Editor. It is also possible to activate the MFE from the FIS Editor, using Edit $\Rightarrow$ Membership Function tab.

The Membership Function Editor makes it possible to perform:

1. determining the universe for each input variable,
2. specifying the number of terms in the universe of the input variable under consideration,
3. determining the shape of terms in the universe of the input variable under consideration,
4. attributing a name to each term of the input variable under consideration,
5. determining the universe for each output variable,
6. specifying the number of terms in the universe of the output variable under consideration,
7. determining the shape of terms in the universe of the output variable under consideration,
8. attributing a name to each term of the output variable under consideration.

The shapes of available terms are compared in Fig. 2.4. Additionally, it is possible to edit each shape. In order to do this, use the special grips available for each shape.

2.3 Rule Editor, Rule Viewer, Surface Viewer

When the following are completed—determining the universe for each variable and attributing the terms (number, shape) to input and output variables—it is then necessary to build a rule base, which will determine the dependencies between the abovementioned variables. This is probably the most difficult operation during the controller design phase. The dependencies mentioned should be formed so as to ensure that while reacting to the change in input variable values, the controller produces adequate changes in output signal values aimed at representing the controlled process accurately.

In this case proper knowledge of the process being controlled is a priority. The person laying out the rule base has to know the process controlled thoroughly, all of its regularities, possible procedures, and reactions to any change in input signal values. Only then, possessing extensive knowledge on possible situations during process control, will it be possible to design the rule base. You should realise that the

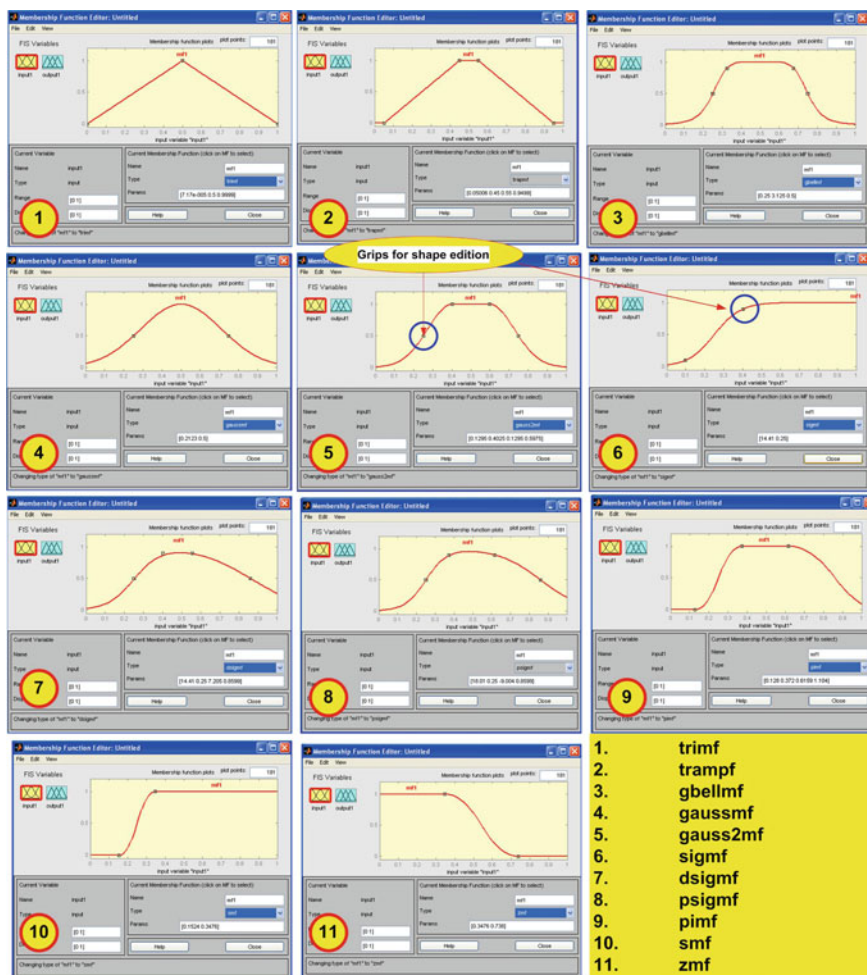


Fig. 2.4 Shapes of available terms in the FIS editor

controller will work strictly according to the pattern given in the rule base, therefore an incorrect layout will result in faulty decisions concerning the process controlled. One of the Fuzzy Inference System elements is the Rule Editor (Fig. 2.5) which allows formulation of the rule base using the entered input and output variables and name recognition of successive terms.

An example rule base is shown in Fig. 2.6. The following condition is pointed out in the drawing:

$$\text{IF}[(\text{stan is mf9}) \text{ and } (\text{doplyw is 1100})] \text{ then } ((\text{elektrownia is duzy}) \text{ and } (\text{upust is 1100})) \quad (2.1)$$

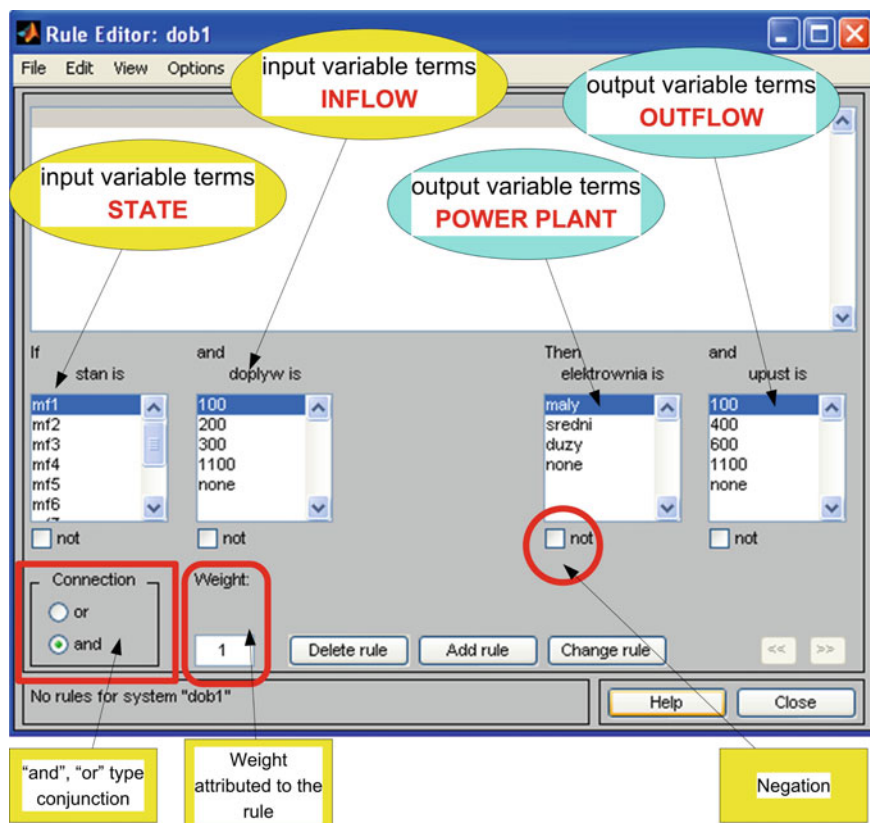


Fig. 2.5 Shapes of available terms in the FIS editor

A very useful tool is the Surface Viewer, that is, the decisive area “visualiser” (Fig. 2.7). The decisive area is the result of an implemented rule base. By way of altering rules in the rule base and then using the visualiser it is possible to observe changes in the decisive area where the controller working point will be moving. Further, using the visualiser showing the decisive area, we may observe changes in the controller output signal values resulting from modified values in its input variables. This analysis makes it much easier to adjust the controller to the process controlled thanks to appropriate formation of the controller’s decisive area, which in turn results from entering possible corrections in the rule base. In the case of one input and output variable, the decisive area is reduced to the controller working point curve (Fig. 2.8).

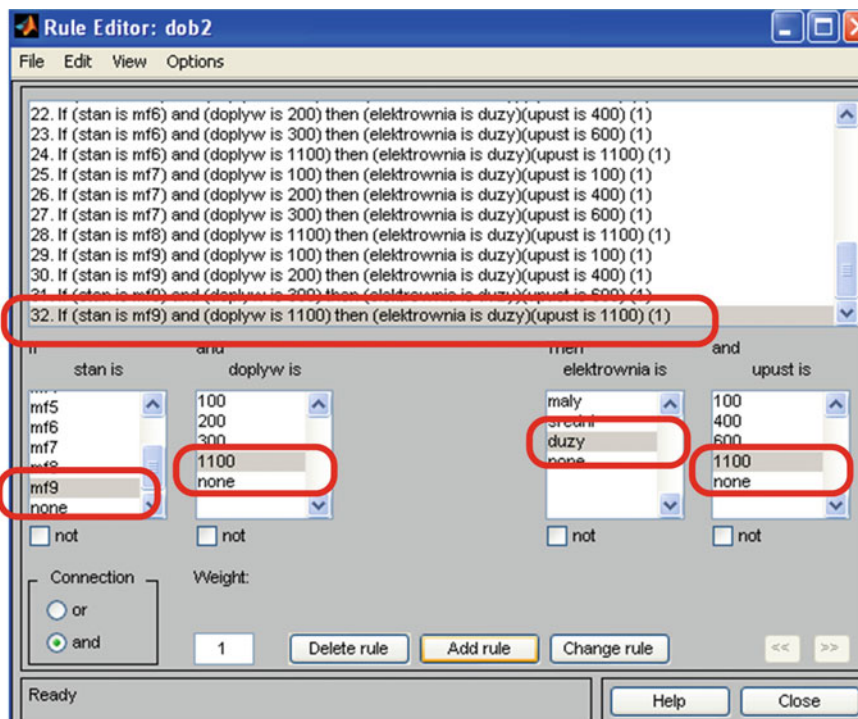


Fig. 2.6 A sample data base

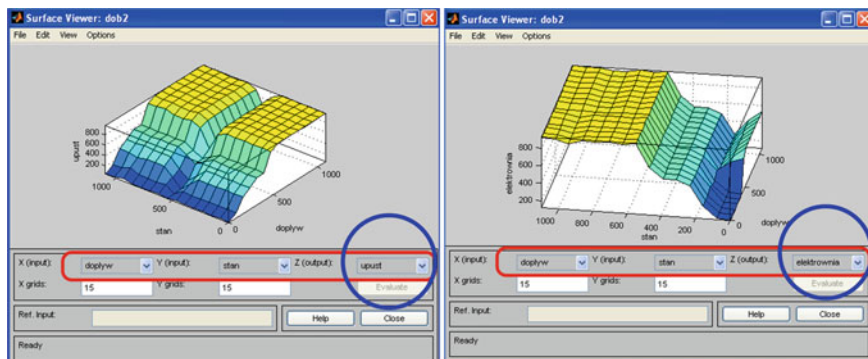


Fig. 2.7 Visualisation of the rule base using the Surface Viewer

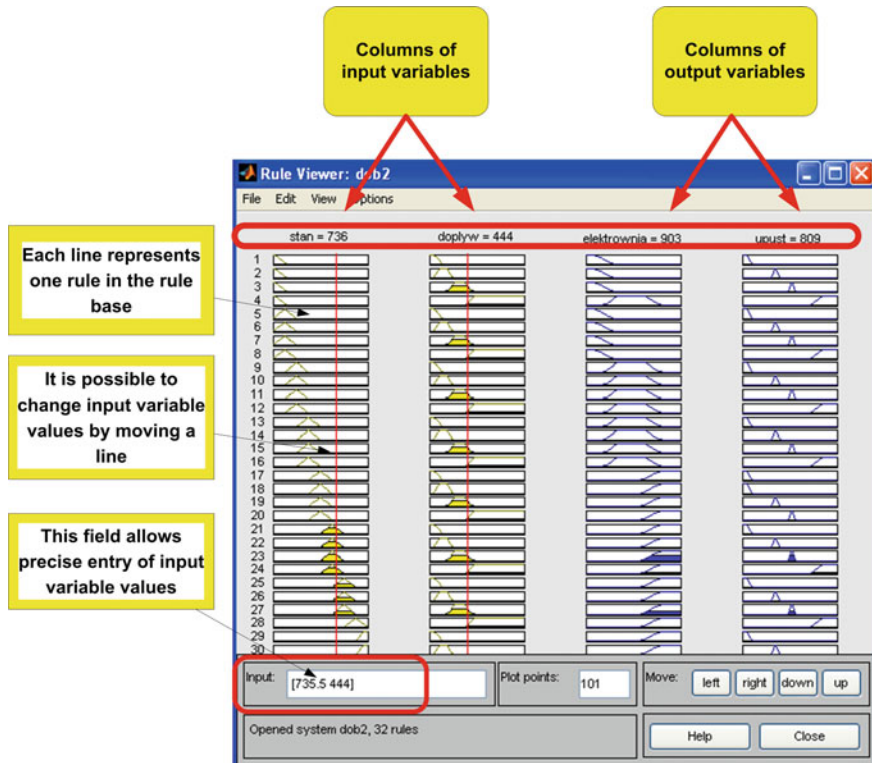


Fig. 2.8 Visualisation of system operation using the Rule Viewer

2.4 ANFIS Adaptive Neuro-Fuzzy Inference System

When using the ANFIS, it is possible to define the FIS object model reproducing dependencies between the abovementioned signals with a certain error. This is carried out with historical input and output data from an observed object characterised by an unknown relation between output signal and input signals.

In MATLAB, in order to create an FIS model in the Takagi-Sugeno structure while using the ANFIS system, it is necessary to define its initial structure. The number of model inputs is not limited, but there can be only one output. The model is created on the basis of some arbitrarily chosen values:

- regarding input variables, it is necessary to specify the variability range (universe) for each variable, and the type and number of terms in the universe selected for each variable,
- regarding output variable, it is necessary to specify the universe and number of functional relations applicable in that universe. In the case of functional relations,

two solutions are acceptable: linear dependencies or constant values, different for individual universe intervals.

Then, it is necessary to activate the adaptive neural network which learns and selects the FIS model parameters on the basis of historical input data for the object observed and historical output data from the object observed. This model will be reconstructing the behaviour of the object observed (physically existing).

Using an adaptive neural network, the FIS model is tuned: the parameters of the input variable membership function are modified, the rule base is formed, and the dependence function parameters are computed.

Learning by an adaptive neural network may be carried out using a reverse error propagation algorithm with either the highest drop method or the hybrid method. The hybrid method involves simultaneous application of the following two methods:

- the least squares method, and
- the method of reverse error propagation with gradient.

The least squares method is used as part of the forward pass—the forward computations, it estimates the conclusion layer parameters—parameters of the $wy = f(we)$ function. Whereas, the gradient method is used during the error propagation phase—*backward pass* and it selects the condition layer parameters and membership function parameters. As a consequence, on the basis of the historical input/output data for an observed object, we obtain an FIS model which simulates the observed object behaviour so as to ensure that the error between output data for an observed object and output data from the FIS model is minimal. Usually, the next step involves checking the operation of the FIS model obtained for a different set of historical input/output data for the observed object, in order to compare the results received in both cycles—training and test. If differences in the results are unacceptable, it is possible to modify the FIS model structure regarding the quantity and shape of terms in the universes of individual input variables, and to have a new configuration of functional relations with reference to the output variable. Then, the FIS model tuning procedure is repeated, and its operation is checked for a test set of input/output data.

2.4.1 The ANFIS Editor

After Matlab application startup, the ANFIS editor is called by entering the *anfisedit*, command. As a result of this, the editor appears on the screen in the form shown in Fig. 2.9.

The list of successively introduced operations will be demonstrated in order to illustrate the whole process involving construction of the FIS (Takagi-Sugeno) model on the basis of an input/output data set for the observed object. We will take a simple dynamical system with one input variable and one output variable (Fig. 2.10)

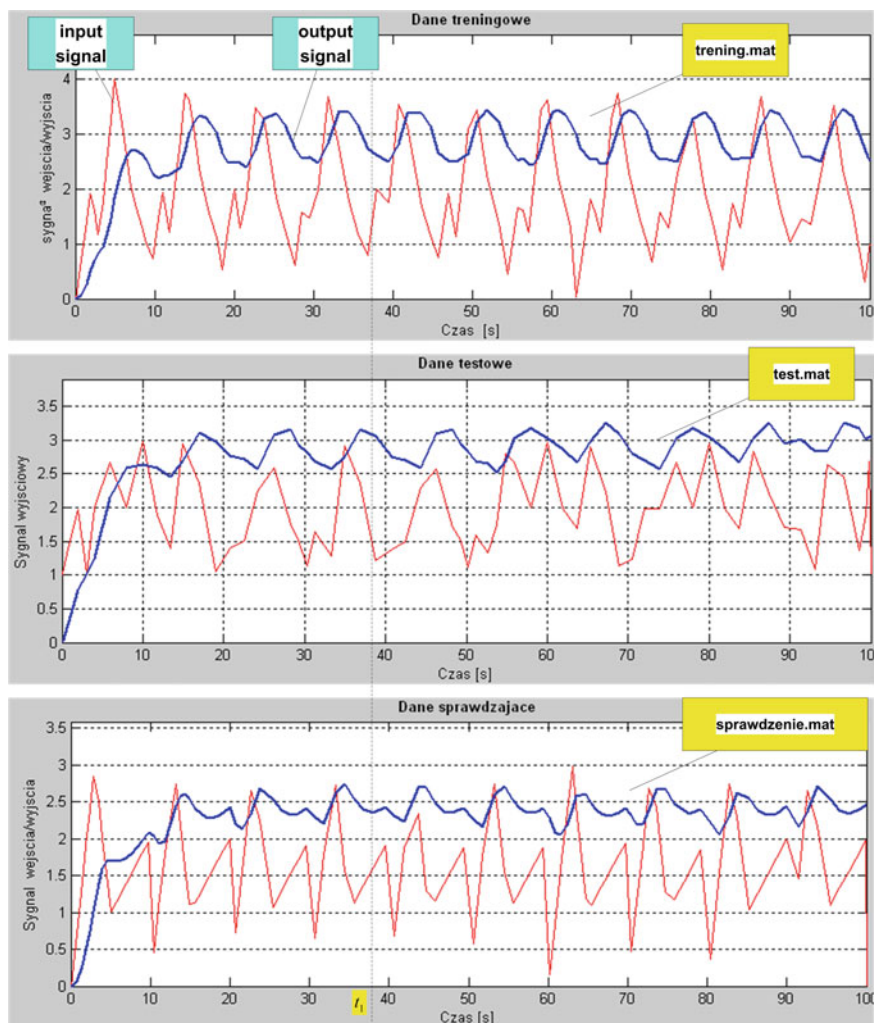


Fig. 2.11 Sets of training, testing and checking data

which will allow us generate a training (*trening.mat*), test (*test.mat*) and checking (*sprawdzenie.mat*) system for the input/output data (Fig. 2.11).

- Then, it is necessary to load to the editor training data (*trening.mat*) corresponding to the first set of historical input/output data for the object to be modelled. In the *Load Data* field, activate the ratio pushbutton by switching it to “Training”, and

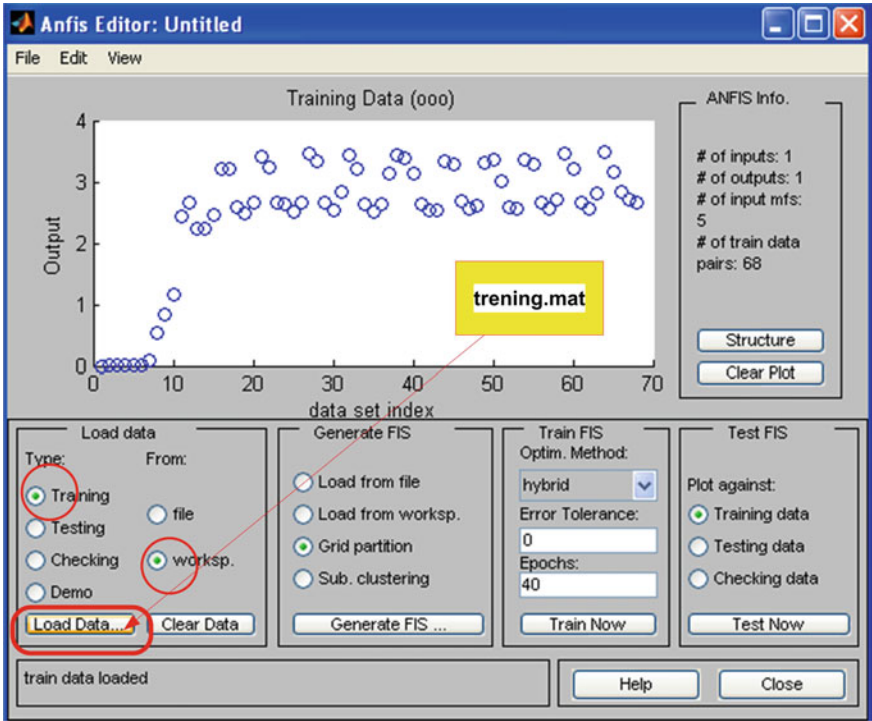


Fig. 2.12 Loading of training data set

- then *ratio* pushbutton to allow loading of the data from disk set “*disk*” or from the Matlab working area “*workspace*”. The editor field called “*Plot region*” will show a diagram presenting the response of the modelled object (historical output data) in the function of historical input data (Fig. 2.12).
- The next step is to generate the FIS (Takagi-Sugeno) structure by specifying the number and shape of terms in the input variable universe, and to declare the choice of linear or constant value of the dependence between the FIS model output and input. As regards the ANIFS editor, restrictions appear in this case which involve an obligatorily assumed number of output functions or constant values corresponding to the approved number of terms for the input variable (input variables). For example, if we take 5 terms (Fig. 2.13) in the input variable universe, this will result in setting 5 functional relations (or constant values) in the output variable universe (Fig. 2.14). If for example there are two input variables, each with a number of

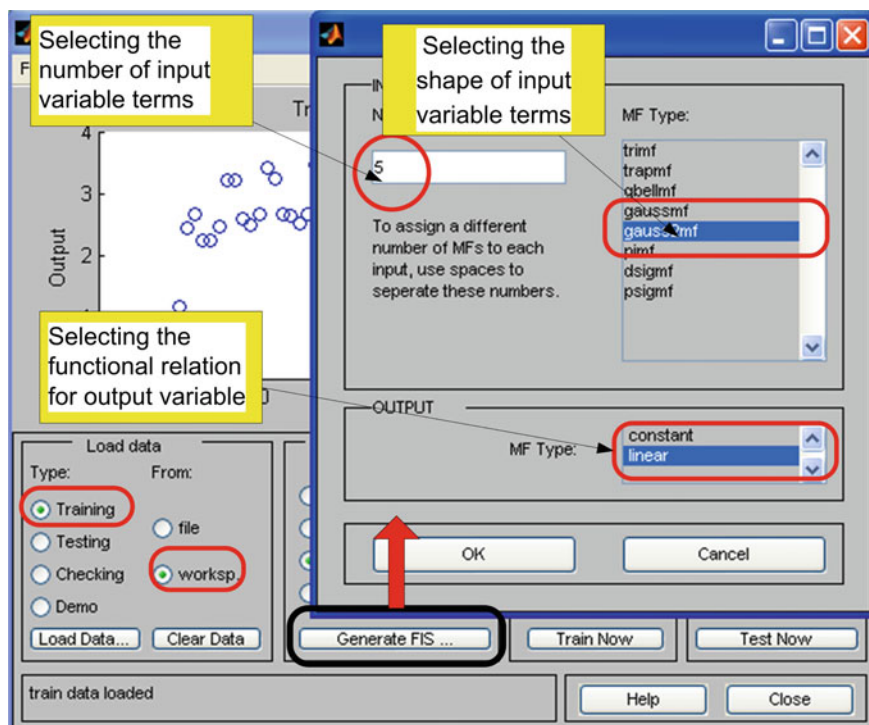


Fig. 2.13 Generation of the FIS structure according to user choice

terms, say 3 and 7 respectively, the number of functional relations (or constant values) for the output is equal (Fig. 2.15). If we take three input variables with number of terms e.g. 3, 5 and 4 respectively, the number of functional relations (or constant values) in the output variable will be fixed at $3 * 5 * 4 = 60$, (Fig. 2.16), etc.

- This step involves activating an adaptive neural network. The FIS model tuning will be the outcome of its operation. Tuning will mean positioning terms in the universe of the input variable and selecting functional parameters for the relationship between output and input with reference to an output variable. The superior purpose of tuning is to reach settings of the FIS model parameters that will make it represent with minimal error the relationship between output and input contained in the training data obtained from the dynamic model (Fig. 2.10). Prior to commencement of tuning it is necessary to set (*error tolerance*), assume the number of learning steps (*epochs*) and specify the network learning method (*hybrid*,

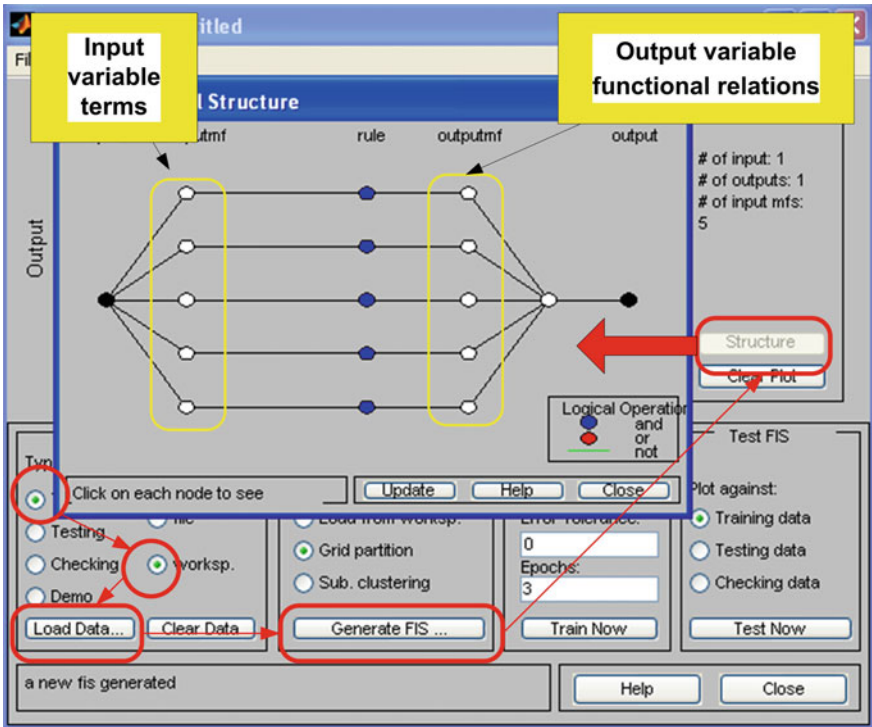


Fig. 2.14 Generated FIS structure according to user choice. One input variable (5 terms)

backpropa). As we mentioned at the beginning, the learning of an adaptive neural network may be carried out using a reverse error propagation algorithm with either the highest drop method or the hybrid method, which constitutes simultaneous application of two methods: the least squares method and the method of reverse error propagation with gradient. After activating the adaptive neural network (in the FIS Test field-*ratio* pushbutton at *Training Data*, then *Test Now*), in the editor field "*region plot*", we observe the error trajectory resulting from the cyclic computations which in consequence lead to determining the FIS model parameters (Fig. 2.17).

Figure 2.18 shows the positioning of terms in the input variable universe, which results from the FIS tuning by the neural network. There are clearly visible changes in both position and shape of terms compared to the original settings (Fig. 2.13). Figure 2.19 presents the rule base resulting from the approved FIS structure (Fig. 2.14). Figure 2.20 is the most interesting. This shows the combination of

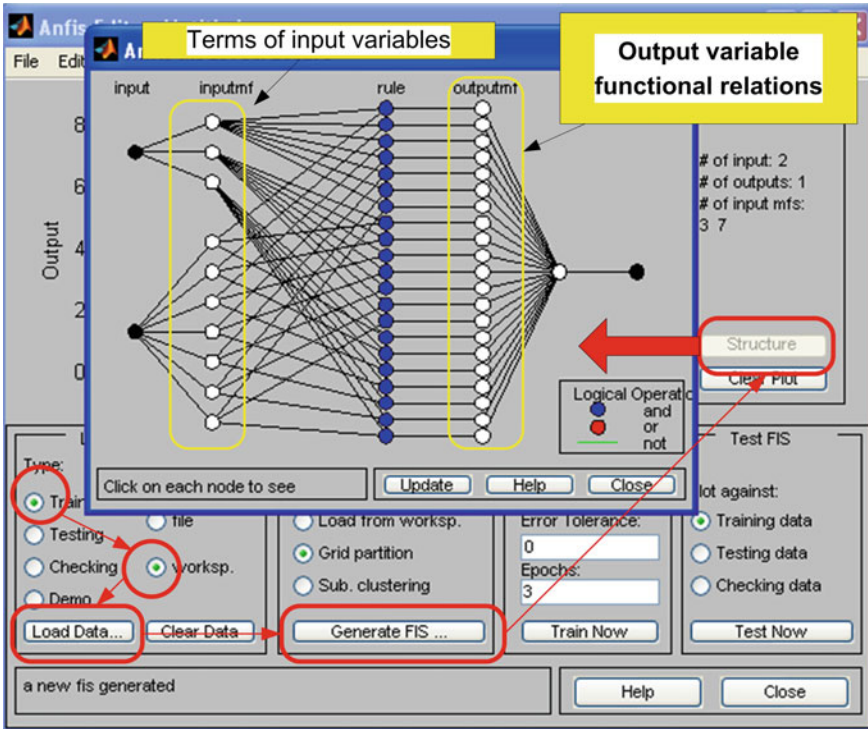


Fig. 2.15 The FIS structure. Two input variables; 3 and 7 terms

declared functional relations defining the dependence between the FIS model output and input. The visible combination is a result of the FIS adjustment by the neural network. As is shown, the FIS adjustment applies both to the distribution and the modification of the input variable terms and forms the FIS input/output characteristic so as to ensure that the error between output training data and output data from the FIS model is minimal. The trajectories of the abovementioned data are shown in Fig. 2.21.

- The next step is to test the FIS model created using the test data set (*test.mat*). In order to do this it is necessary to load the data into the editor. In the *Load Data* field, set the ratio pushbutton at “*Test*” and then at a position allowing the data to be loaded from the disk set “*disk*” or from the Matlab working area “*workspace*”. The editor field “*Plot region*” will show a diagram presenting the response of the modelled object according to the historical input data (Fig. 2.12). Then, in the *FIS*

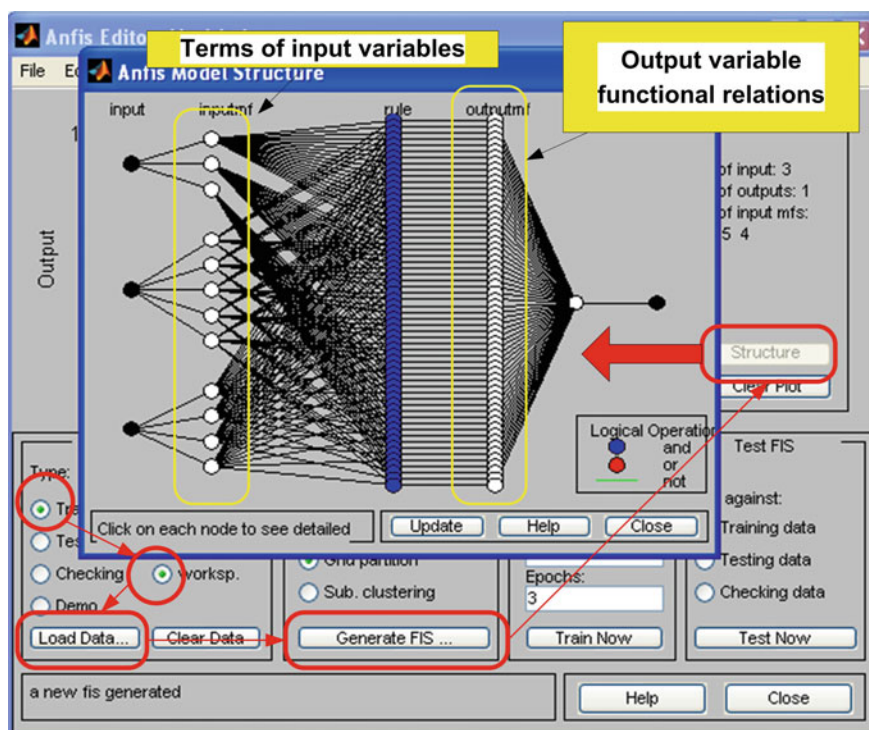


Fig. 2.16 The FIS structure. Three input variables; 3, 5, 4 terms for each variable

Test field, set the *ratio* pushbutton to *Testing data* and press the *Test Now*. As a result, the “*Plot region*” field will display a second diagram demonstrating the response of the created and tuned FIS model to the same input data (Fig. 2.22).

- The final step involves checking the created FIS model using the check data set (*sprawdzenie.mat*). As before, in order to do this it is necessary to load the data into the editor. In the *Load Data* field, set the ratio pushbutton at “*Checking*”, and then at a position allowing the data to be loaded from the disk set “*disk*” or from the Matlab working area “*workspace*”. The editor field “*Plot region*” will show a diagram presenting the response of the modelled object to the historical input data (Fig. 2.12). Then, in the FIS Test field, set the ratio pushbutton to *Checking data* and press the *Test Now*. As a result, the “*Plot region*” field will display a second diagram demonstrating the response of the created and tuned FIS model to the same input data (Fig. 2.23).

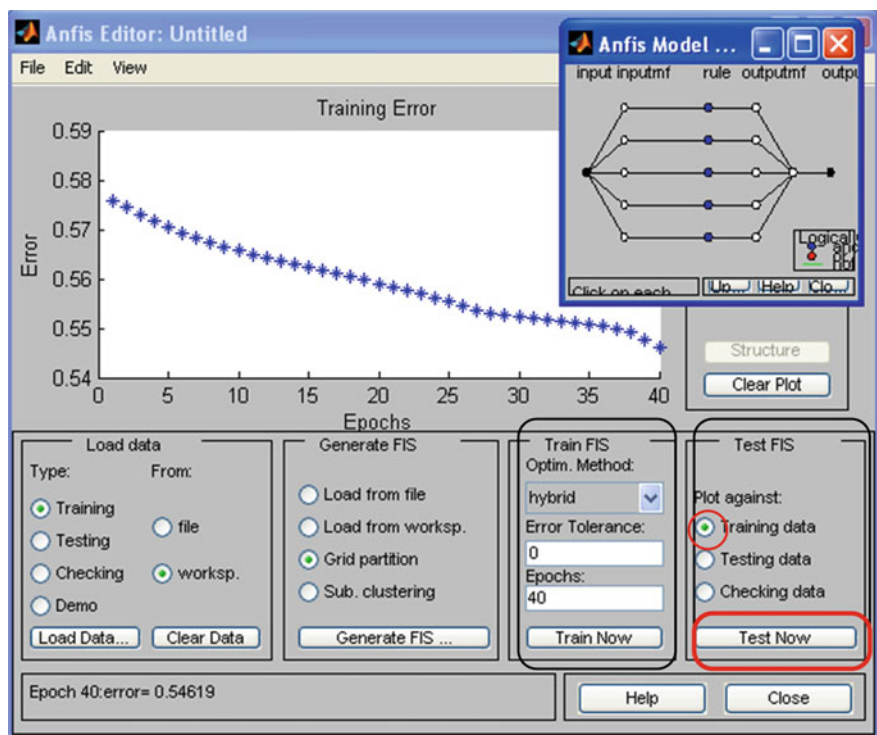


Fig. 2.17 Error trajectory

As we may observe, the trajectories of the output test data and the output data from the FIS model differ considerably, while the general tendency in changes and variability range remain the same. In the numerical example demonstrated, the output signal changes as a function of input signal changes are very substantial (Fig. 2.11). This enables us to conclude that the assumed FIS structure (Fig. 2.14) is insufficiently accurate to capture rapid changes in output signals in relation to input signals. In this case, we should experiment with other FIS structures, aimed at making both diagrams shown in Fig. 2.23 as close as possible.

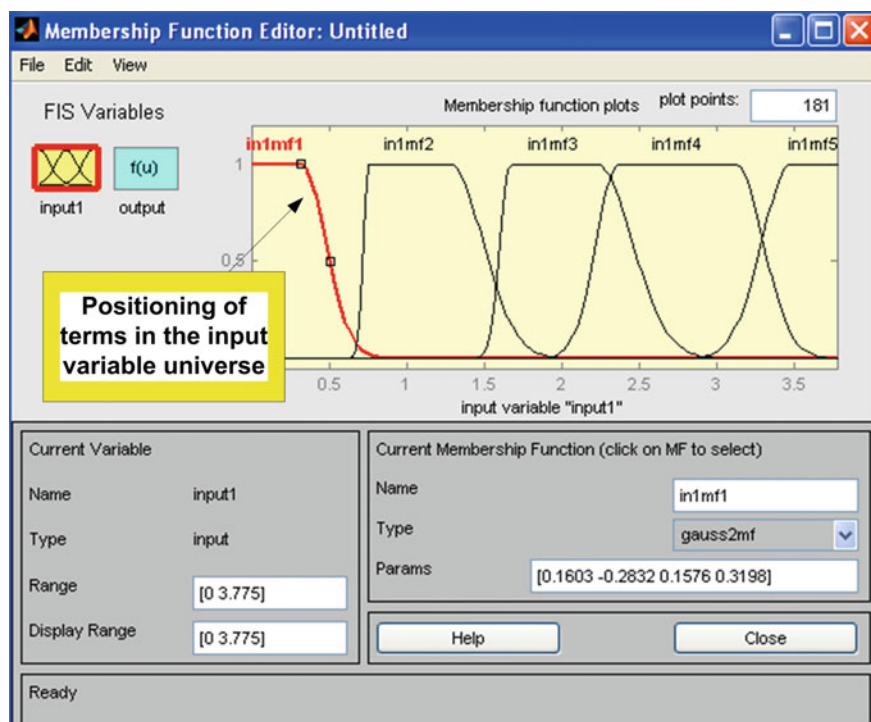


Fig. 2.18 Position of terms in the input variable universe defined as a result of neural network operation

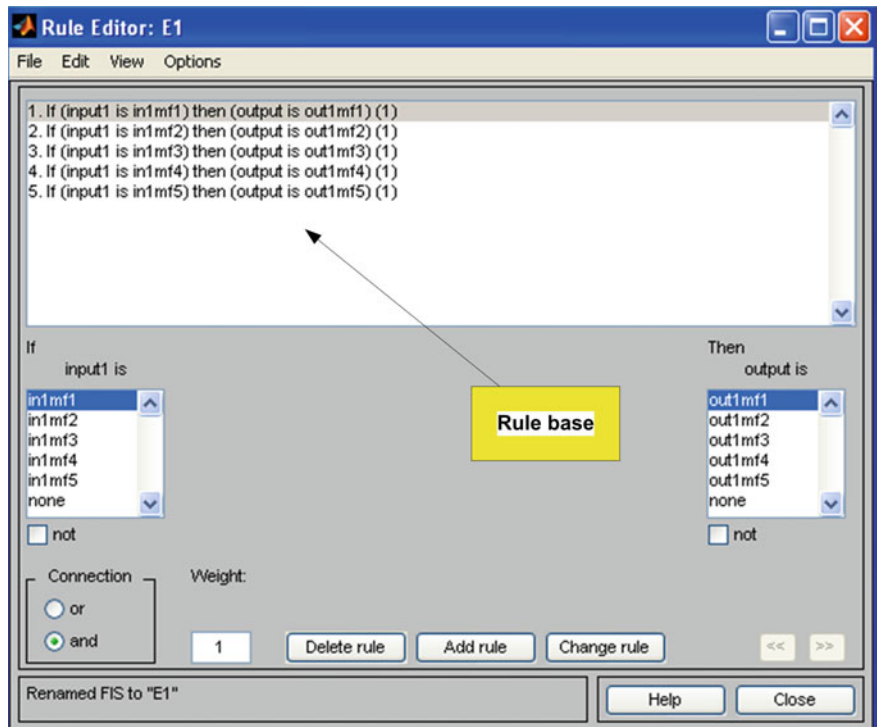


Fig. 2.19 Rule base generated by the ANFIS

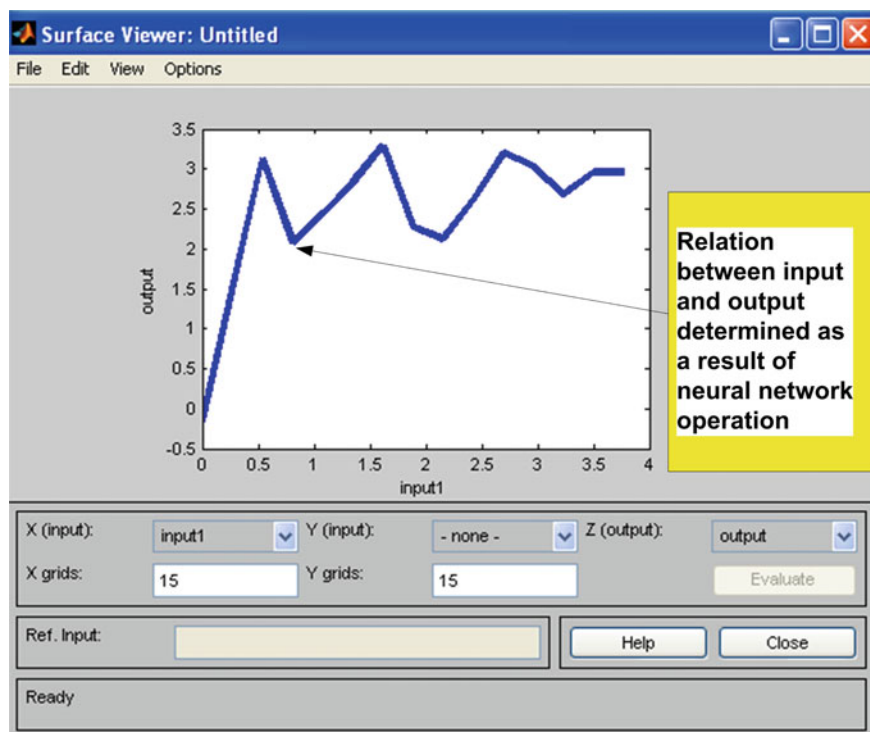


Fig. 2.20 Relationship between input and output defined as a result of neural network operation

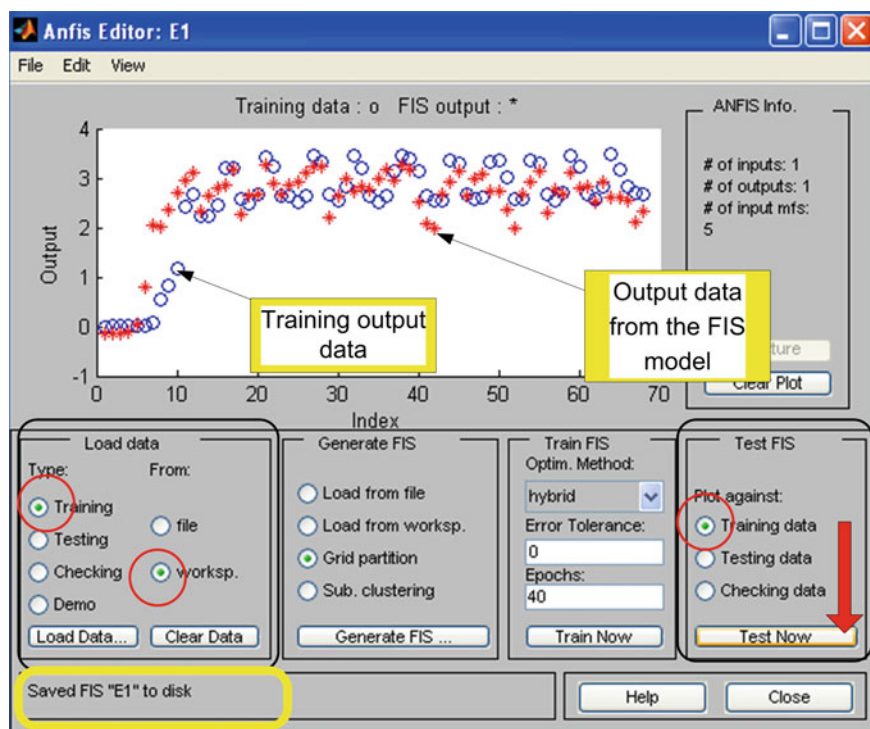


Fig. 2.21 Trajectories of output training data and output data from the FIS model

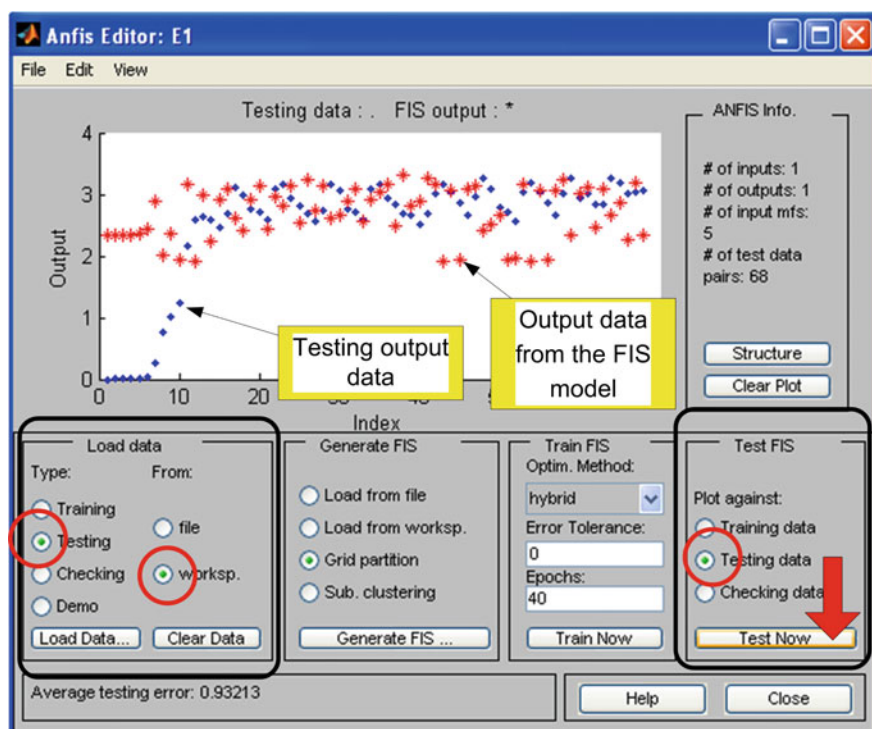


Fig. 2.22 Trajectories of output testing data and output data from the FIS model

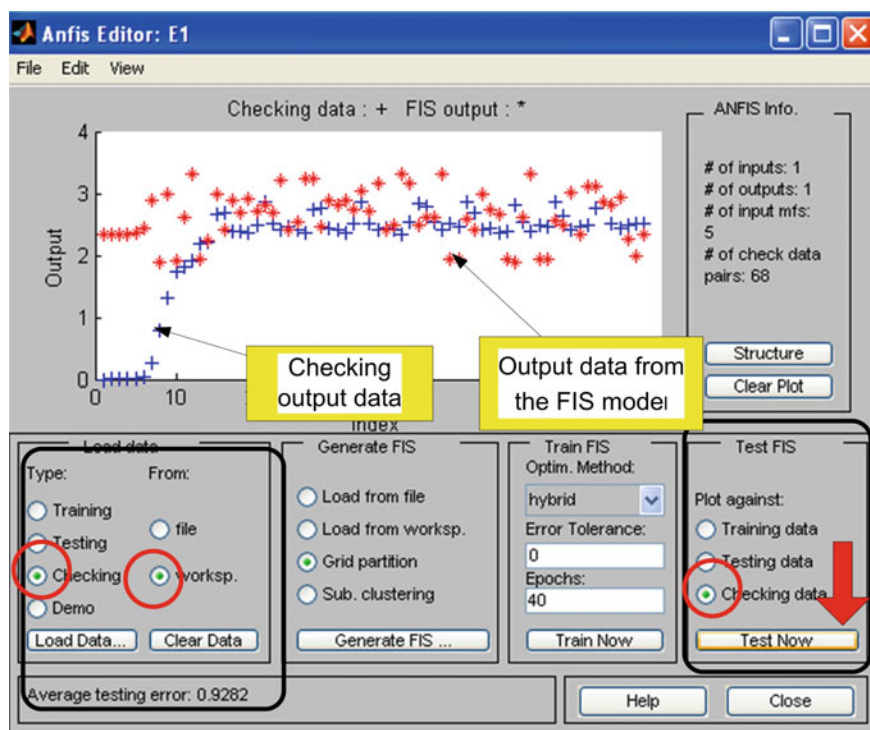


Fig. 2.23 Trajectories of output checking data and output data from the FIS model

Fuzzy Control in Environmental Engineering

Chmielowski, W.

2016, XXVIII, 276 p. 258 illus., 250 illus. in color.,

Hardcover

ISBN: 978-3-319-19260-4