LFLCSim: Program for simulation LFLC

1 Introduction and main information

LFLCSim is a simulation program, which is effectively used for a control in a close loop (see [4] for more detail). LFLC (Linguistic Fuzzy Logic Controller) can be of type P, PI, PD or PID and its linguistic description must be prepared by a program LFLC2000. The process is assumed to be described by an ordinary differential equation. You can modify various process and control parameters including the linguistic description with help of LFLCSim.

According to the type of fuzzy controller you must enter linguistic context of error (or change of error and difference of error change) and linguistic context of a control action (or change of control action), before you start a simulation. You can choose from the four possibilities:

1. no change of automatic context,
2. automatic context, which is changed during simulation,
3. no change of own context,
4. own context, which is changed during simulation.

These possibilities will be explained later. At this point we only note that the concept of the linguistic context allows to simplify fuzzy controller project and to make it more effective.

The main function of the program LFLCSim is to establish and to display output response of simulation process during its control with the help of fuzzy controller. It is generated by repeated click of the keys **Enter** or **CtrlEnter**. We can also change setpoint value — a value, which we want to reach — we can add random disturbances, display fired rules during the simulation and particularly watch step by step control actions and rules, which were fired. By these observations you can immediately modify the linguistic description to get optimal result. Thus, both programs LFLC2000 and LFLCSim become a strong tools for a quick and effective project of fuzzy controller.

The following symbols are often used in LFLCSim program:

E	*error*,
dE	*change of error*,
$d2E$	*difference of change of error*,
U	*control action*,
dU	*change of control action.*

2 Functions of the program LFLCSim

2.1 File

When you select this item, you have the following possibilities:

New Project, *Open Project*, *Save Project*, *Save Project As*, *Close Project*, *Open Description* and *Exit*.

The setting of all important values, e.g. constants of the selected differential equation, start conditions, name of the linguistic description, type of fuzzy control, setpoint value and others, is called a project. We start working with this program by choosing item *New Project* or opening of already prepared linguistic description (*Open Description*). For maximal comfort of users with program LFLCSim we can *Save Project* under a name and *Open Project* if it is needed. Thus the program is immediately prepared to work. Observe the project file is added the extension ".prj" and its linguistic description ".rb". For proper closing of a project we select *Close Project* and for closing of a program LFLCSim we use *Exit*.

2.2 Settings

When we click on *Settings* and choose whichever from the offered items, a window opens, in which global information about the concrete simulated process are specified. The window has the following Tab-pages:

- Project & Description
- Process Settings
- Inference Settings
- Simulation Context

- Response Context

2.2.1 Project & Description

In this window, the user see the project name and all path of the saved project if it has already been specified. If the linguistic description has already been opened, its file name and its description comment is also displayed.

The last thing, which we find in this Tab-page, is *General Settings.* Item *Use COM Object* is used to read .rb descriptions and for fuzzy inference. If we mark *Derivates* item, we will work with derivates of the errors $(dE, d2E)$. Further, we can define the sign of the error E. There are two possibilities:

$$E = v - y$$

and

$$E = y - v,$$

where y is the output value of the process and v is the setpoint value.

2.2.2 Process Settings

To simulate the control, first, we must specify a process, which we want to control. It is implemented through the ordinary differential equation with constant coefficients, which characterize the process behavior. But then we must enter its coefficients and possibly also some nonlinearities. The selected equation of the controlled process is displayed at the bottom of the screen. Note, that a stable linear process of the first order is predefined.

We can choose process of the first, the second or the third order, as we can see in figure 1. As soon as we choose some of them, we can enter its coefficients, a time delay (*Delay T*) and a sample period (*Sample T*). Besides, we can also enter some nonlinearities by click on **Nonlinearities** button. Then the menu with the list of possible nonlinearities opens and they can be added to the selected differential equation. Each nonlinearity can be multiplied by any coefficient f. Number of the nonlinearities is limited to five in one process.

2.2.3 Inference Settings

In this Tab-page, we can choose type of fuzzy inference. It means specification of type of says, how fuzzy IF-THEN rules should be explicated and what kind

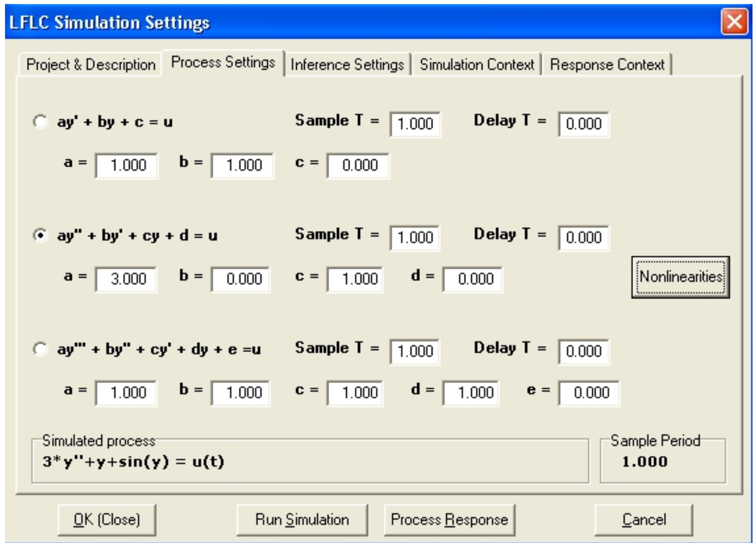

Figure 1: Choice of process type

of approximate reasoning method should be used. The following options are available:

- Perception-based Logical Deduction
- Fuzzy Approximation (CNF)
- Fuzzy Approximation (DNF)

Moreover, we can specified the conjunction type of the DNF approximation. We can choose minimum, product or Lukasiewicz type.

Next possibility is to specify defuzzification method, i.e. to determine the way of defuzzification of the fuzzy set obtained as a result of approximate reasoning. We can choose from the following options:

- Defuzzification of Evaluative Expression
- Simple Defuzzification of Evaluative Expression
- Simple Center of Gravity
- Modified Center of Gravity

- Mean of Maxima
- Smooth Defuzzification of Evaluative Expression

For more details about types of fuzzy inference and defuzzification methods we refer user to the book [4] or LFLC 2000 Help.

2.2.4 Simulation Context

In the *Simulation Context* Tab-page, first, we have to specify the type of controller by choosing one of the following four options: P, PI, PD or PID controller. Note, that the number of the variables in opened file of linguistic description must correspond to the type of fuzzy controller (i.e. it is needed to have 1 arbitrary variable for P, 2 arbitrary variables for PI and PD controller and 3 arbitrary variables for PID controller). Further information about type of controller can be found in book [4].

When we choose the controller, we must also specified the linguistic context (see Figure 2).

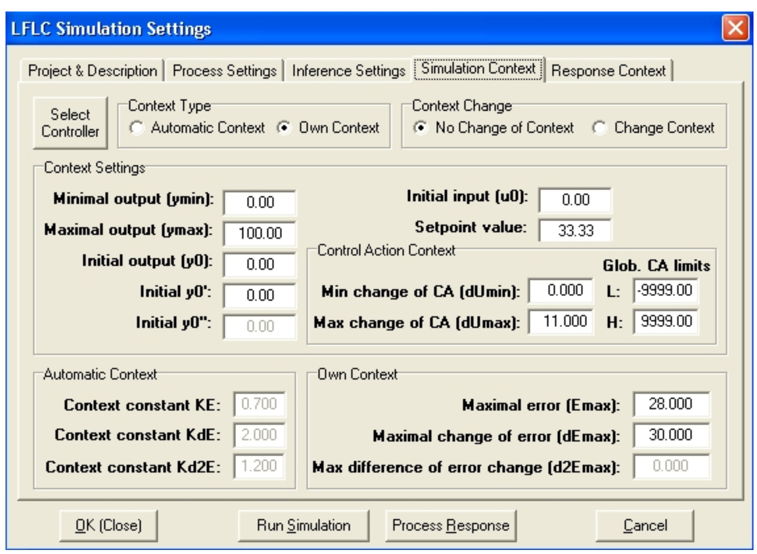

Figure 2: Simulation context Tab-page

As we mentioned at the beginning, we can choose from four possibilities:

1. no change of automatic context,
2. automatic context, which is changed during simulation,
3. no change of own context,
4. own context, which is changed during simulation.

The automatic context means, that LFLC adjusts the context for the error, change of error or difference of error change (if we work with PID controller) automatically. However, the control action context is needed to enter.

The own context means to enter it your own. If we have already opened linguistic description, the context will be automatically used from it.

Furthermore, we can also enter, if we want to change context or do not want to change it during the simulation. The change of context is very interesting possibility how to reach very precise fuzzy control. In fact, it is caused by a contraction of context. However, this problem is not so easy, because we must also take into consideration a possible instance of a big error and a disturbance. There is a special algorithm in LFLC, which implements change of context automatically. In spite of this, we can set all values manually before it changes automatically. This possibility can be turn off during the simulation by click on **Change** button. If we select own context and its change, it is possible to change context in each sample period. If we select automatic context and its change, we can change automatically modified values of the context of all variables only if LFLC "thinks" it should be change.

When we select the way of work with context, we should enter minimal $ymin$ and maximal $ymax$ output values, which are suitable for depicting of the response on the screen. These values should correspond to our particular problem which is simulated.

Next, we enter initial output value y_0, possibly on the base of the process type, also initial values of the first y_0' and the second derivative y_0''.

When we use PI or PID fuzzy controller, we can also enter initial input value u_0. This input is modified during the simulation. When we use P or PD controller, this item is not available because it is possible to simulated it by entering disturbance at the beginning of the control.

We also enter a setpoint value, a desired value on the output.

We can also specify control action values of the context. It depends on the selected type of fuzzy controller. We enter minimal $Umin$ and maximal

Umax possible control action in a case of P and PD fuzzy controller and minimal *dUmin* and maximal *dUmax* possible change of control action in a case of PI and PID fuzzy controller. Moreover, for the last two fuzzy controllers, we also enter limits for minimal and maximal possible (global) control action. It corresponds to a technical stop, because control action can not be increased or decreased without any limits.

If we select automatic context, we can set context constants - *KE* (P controller), moreover *KdE* (PD and PI controller) and on top of that *Kd2E* (PID controller). These constants have the important role because they are used during modification of automatic context. Context of the independent variables "learned" by LFLC has the following values:

$$\begin{aligned} \max(E) &= KE \cdot E_0 \\ \max(dE) &= KdE \cdot \max(E) \\ \max(d2E) &= Kd2E \cdot \max(dE) \end{aligned}$$

where E_0 means the error at the time $t = 0$. The context constants are set to the certain values which were established by the LFLC development. They do not have to be changed. But you can try to make an experiment with them to find out their influence on control process. Generally speaking, the decreasing of *KE* lead to the faster reach of desired value, but it can cause the bigger overshoot. On the other hand, the increasing of *KE* can cause the decreasing of a overshoot and lead to smooth response, but at the same time the speed of controller will be smaller. It is necessary to find a balance between these constants in relation to the control action context, if you decide to change them.

If we select own context, we should also set error context, change of error context and possibly difference of error change context. However, the context is read from LFLC 2000 automatically, thus we do not have to set anything. Note, that the linguistic context is understood here as symmetric.

2.2.5 Response Context

The last Tab-page is related to the *Process Response* button, which is used for generating of the process response (it will be described later). For now note, that this window allows us to set the context values (minimal and maximal output, initial output, initial output of the first and second derivatives, initial input and setpoint value).

2.3 Simulation

After you enter initial and context values at Simulation Context Tab-page, you are prepared to start simulation by click on **Run Simulation** button at the bottom of the screen. Apart from a small simulation control window, the main simulation window will be opened. There is output response of the controller generated, the setpoint value and disturbance are entered, control actions and fired rules are verified. It is possible to change the linguistic description, initial values and context values and also to test the behavior of the description during the simulation.

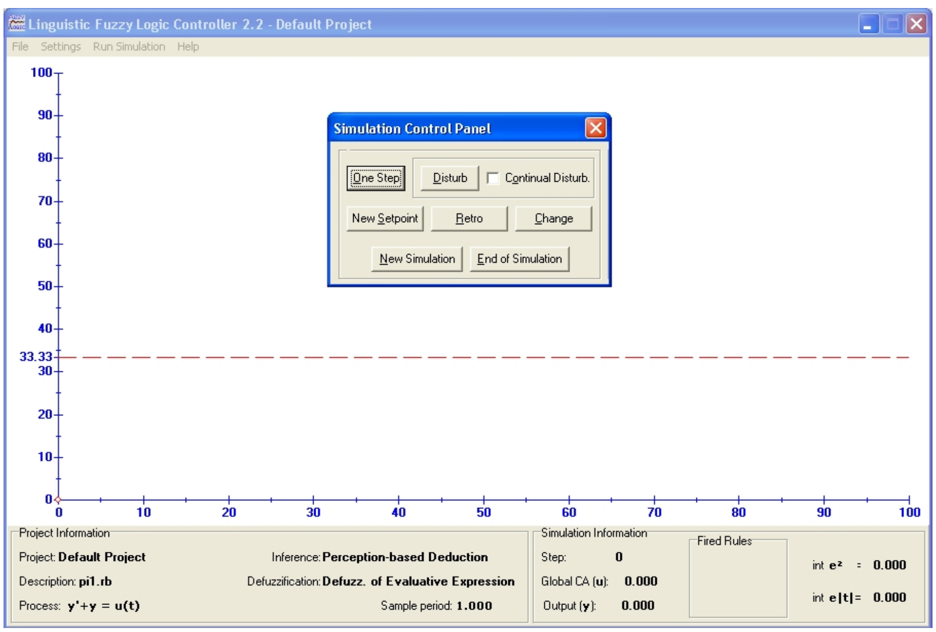

Figure 3: Introductory simulative screen

The introductory simulative screen is depicted in the figure 3. Axis y corresponds to the output values of the process and axis x means the time axis with the marked sample time.

There is a information window under the axis x, where the following values are continuously changed:

step t	step of control action
u	control action
y	output value of the process
$\int e^2, \int e\|t\|$	integral criterions

Integral criterions are normalized on the rectangle $t \times \max(E)$ and it allows the mutual comparison of the different simulations. At each step there are also fired rules depicted.

It is very easy to generate output response. Firstly, we must to enter the setpoint value. Then the output response is generated either by clicking on **One Step** button or pressing Enter key. One click or one press generates the response for one sample time. If we repeats it, step by step we can watch the behavior of our fuzzy controller, immediately find its weak places and modify context as well as the linguistic description.

If some rule was fired, the time sample in output response is depicted by a red circle. In opposite case, it is drawn by a green square. It immediately allows to see the week places of just used linguistic description.

The output response should be generated until the setpoint value will be reached. Remark, that fuzzy controller rarely reaches the setpoint value and often oscillates around stable state. It is a price paid for the other useful properties of the fuzzy controllers. However in practise, it is not possible to require absolute precision, since a real process is influenced by the various disturbances. Therefore fuzzy controllers are optimal compared to the classical ones. The oscillations can be eliminated by a change of context during the simulation. Moreover, the setpoint value can be reached quite *precisely* (provided that there are no random disturbances generated).

We can change the setpoint value (**New Setpoint** button) and establish the disturbances (**Disturb** button) at any moment of the generated output response. **Disturb** button establishes only one disturbance. The disturbance is then at the following time sample set to zero. If we want to generate the sequence of random disturbances we mark *Continual Disturb*. Then we can enter the value range which the random disturbances will be taken from.

If we want to repeat the simulation with the same context and the initial values, we click on **New Simulation** button. Of course, we can enter new setpoint value and also new disturbances if we want.

The **Change** button we use, if we want to change state, when context is changed during the control and when not. It means that the change context values is either offered us or not. It is possible to enter new context values by standard editing keys or left click of the mouse. Repeated click on this button switches on and switches off the possibility of the another modification of the change context.

If we want to enter a new context, we have to close the small simulation control window and choose *Settings* → *Simulation Context*. The context values can be changed there and then we can repeat the simulation again.

If we want to modify the linguistic description, we have to open corresponding .rb file in a program LFLC2000, in which existing rules can be deactivated or new rules can be added (more information about this problems can be find in LFLC 2000 Help). If these changes were saved in LFLC2000, we will be asked to reload the modified linguistic description and to load the context of variables after we choose *Run Simulation* → *Fuzzy Control*. The simulation with the new linguistic description can be then repeated.

We can also find out detailed information about behavior of the control process by a click on **Retro** button. A new window with the important values for a given sample time and a blue circle on a depicted curve of response appears (see figure 4). By clicking on the arrow buttons we can see, how the blue circle moves on the depicted curve and how the various values of the process changes.

At every single time sample t, we can thus observe error E_{t-1}, change of error dE_{t-1}, difference of error change $d2E_{t-1}$ (in case of PID control), control action U_{t-1}, possibly change of control action dU_{t-1} and output value of the process y_t. Possible disturbance, context values and numbers of fired rules are also pictured. Notice, that the values E_{t-1}, dE_{t-1}, $d2E_{t-1}$, dU_{t-1}, U_{t-1} correspond to the sample time $t-1$, which precedes t. Similarly, also context values and fired rules correspond to the previous time sample. It means, that we get information about values, rules and control action, which led to the final output value y_t at the time moment t.

We can also look at the form of fired rules by a click on **Show Rules** button. This window can stay opened and at the same time we can use arrow buttons. When the cursor on the depicted curve moves, both windows are updated. This window can be closed by a click on **Hide Rules** button.

We can store the information about behavior of the control process into a text file, which can be used for a subsequent analysis. First of all we set the file name (**Set File Name** button) and then by click on **Store Values** button the values of process will be stored into the setting file.

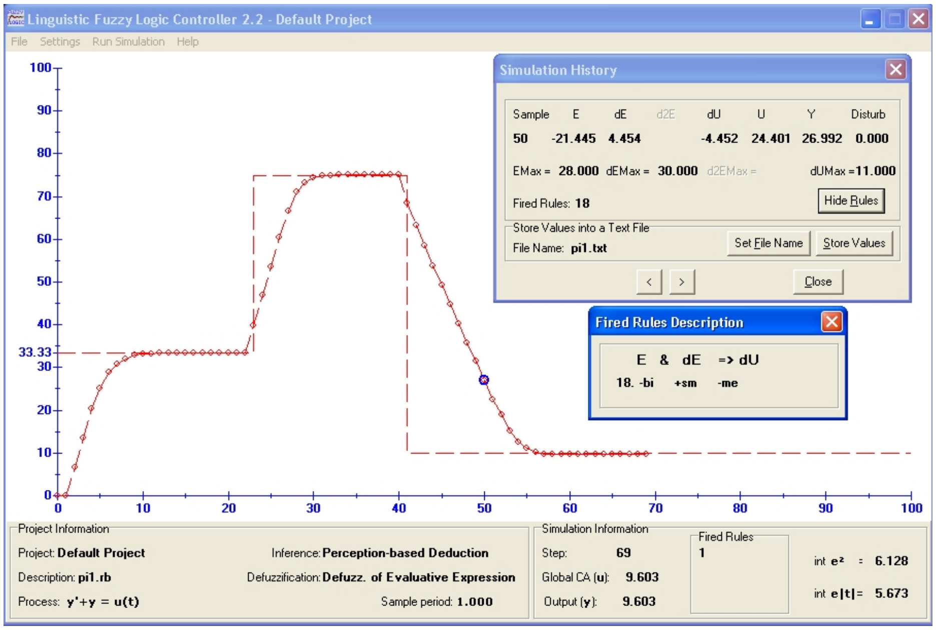

Figure 4: Simulation screen with retrospective of the control process

2.4 Process Response

Process response is an auxiliary function, which allows to generate process response to impuls, jump or another input signal. We can use it to learn about the behavior of a controlled process and to set suitable context. This function is similar to simulation described above but it is more simpler. The result of this function is the response of the simulated process.

After choosing *Run Simulation* → *Process Response*, the response context can be defined (see Figure 5). Here we can set minimal *ymin*, maximal *ymax* and initial output value y_0, initial input value u_0 and setpoint value.

Depending on the order of the used differential equation we can also set initial value of the first y_0' and the second derivative y_0''. The values *ymin* and *ymax* are used for the right drawing of the output and they would correspond to the concrete problem, which is simulated.

We start it up by clicking on **Process Response** button which is situated at the bottom of the simulation settings window. A main screen which

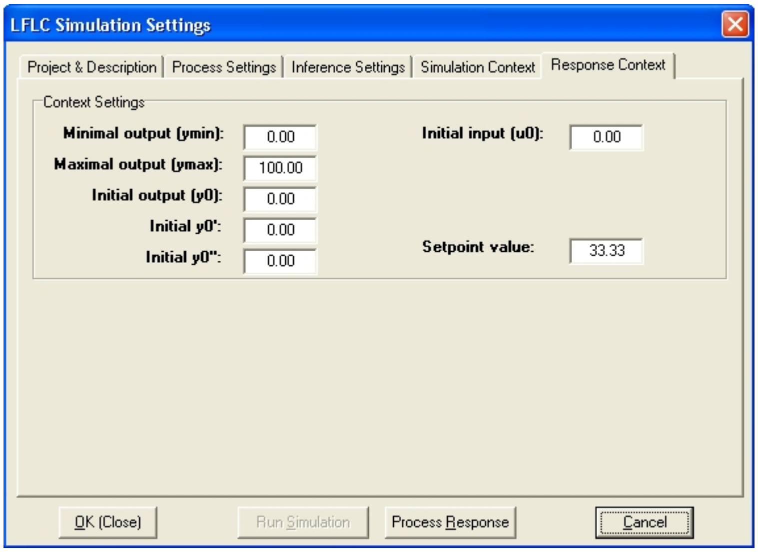

Figure 5: Response Context

is similar to the screen for simulation, appears (Figure 6). There is the process response generated. The largest part of the screen is devoted to the coordinate axes. Axis y corresponds to the output value of the process and axis x is time axis divided into the time samples. There are the same information and values displayed as in the main simulation window below the axis x.

Before generating the response, we must set the input value (**Set Input**). Then the process response is generated by repeated pressing of Enter key or clicking on **One Step** button. Each press or click generates one time sample and each click on **10 Steps** generates ten time samples. If we want to set input for each simulation step, we mark *Continuous Input* and if we want to set the input change for each simulation step, we mark *Input Change Cont.* After we click on **One Step** button, we are ask for the setting the input value or input change value. Each step can be returned by **Step Back** button. If we want to change the setpoint value, we click on **New Setpoint** button and set the desired value. If we want to generate a new response, we use **End of Process** button. Introductory screen opens and procedure above can be repeated.

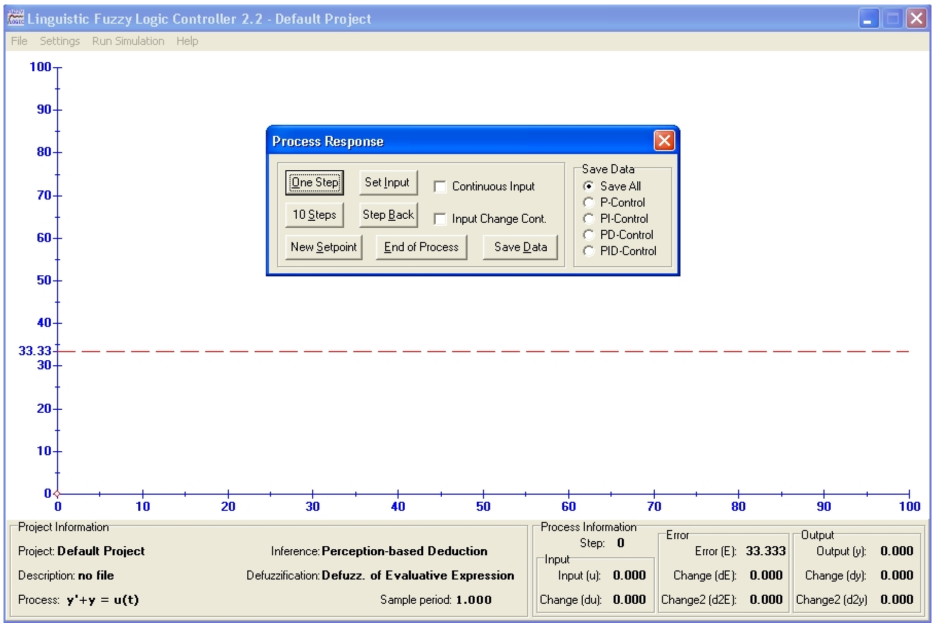

Figure 6: Introductory process response screen

It is possible to save information about behavior of the process response into a text file and use them for a subsequent analysis (these data are often used for learning rules in LFLC2000). At first we have to decide which data should be saved. It depends on the type of control, which was thought (*P-Control, PD-Control, PI-Control or PID-Control*). We can also save all data (*Save All*). Next we click on **Save Data** button and we set the file name.

This file will have the following structure:

Item	**Format**	**Comment**
Step	3	
Error E	10.039	All controls
Change of error dE	-3.161	PI, PD and PID fuzzy control
Difference of error change $d2E$	-3.161	Only PID fuzzy control
Control action u	8.000	P and PD fuzzy control
Change of control action du	3.000	PI and PID fuzzy control
Process response y	6.220	
Change of response dy	3.059	
Difference of response change $d2y$	-0.102	
Setpoint value v	13.200	

At the end we recommend the following literature, which can be helpful in understanding this software:

References

[1] V. Novák, Fuzzy množiny a jejich aplikace. SNTL, Praha 1986 and 1990. (Fuzzy Sets and Their applications; in Czech)

[2] V. Novák, Fuzzy Sets and Their Applications. Adam Hilger, Bristol 1989.

[3] V. Novák, I. Perfilieva, and J. Močkoř, Mathematical Principles of Fuzzy Logic. Kluwer, Boston/Dordrecht 1999.

[4] V. Novák, Základy fuzzy modelování. BEN, Praha 2000. (Foundations of fuzzy modeling; in Czech)