7

Design of fuzzy systems using the fuzzy shell FLOP

Overview

The fuzzy shell FLOP (**F**uzzy **L**ogic **O**perating **P**rogram) allows the design and the analysis of rule based systems on the basis of fuzzy logic. The program offers the following options:

- definition of linguistic variables and corresponding terms
- creation of rule bases
- realization of inference processes
- evaluation of transfer characteristic curves and maps
- simulation based on record data
- creation of fuzzy controller files for the blockoriented simulation system BORIS
- DDE interface to other applications

Various operators, inference mechanisms and methods of defuzzification are selectable for the different arithmetic operations which are generally be represented graphically. Triangle, trapezium and singleton are possible for the type of membership functions.

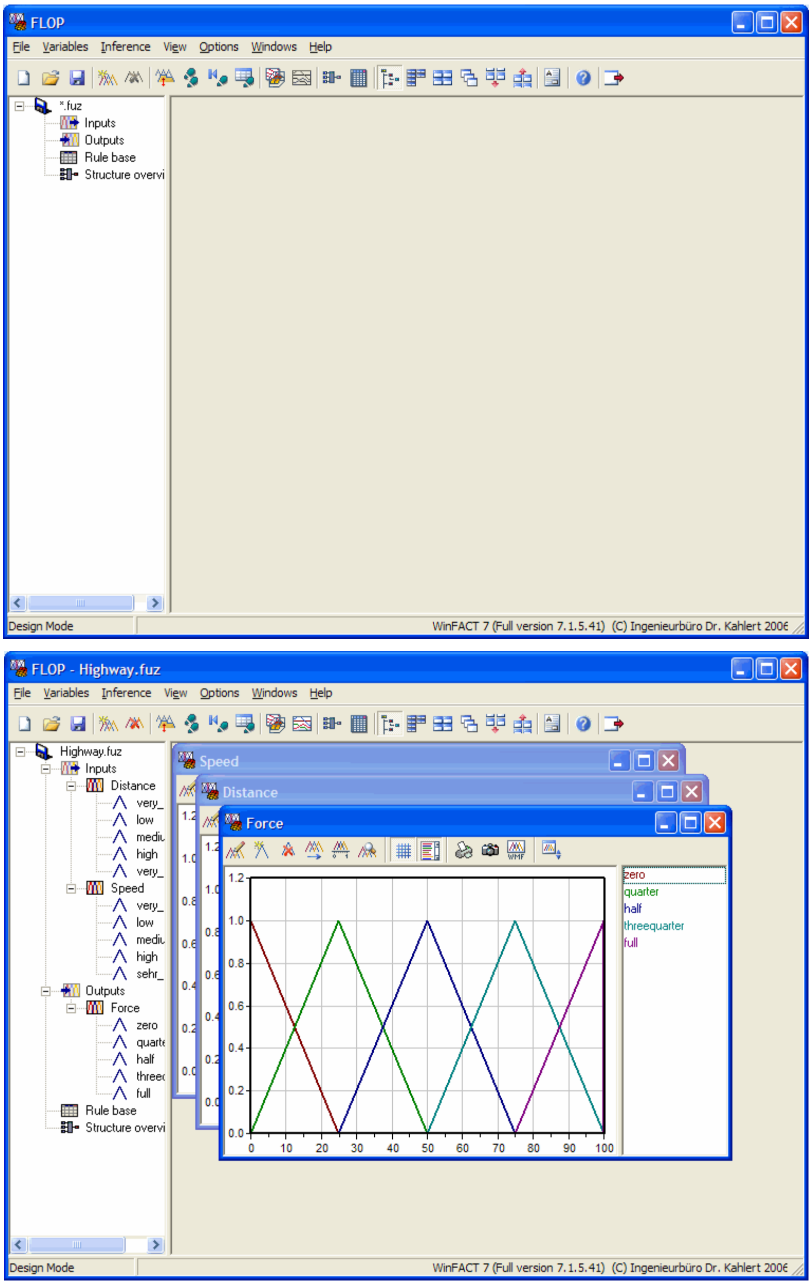
A complete introduction into the basics of fuzzy logic and possible fields of its application can naturally not be given here. Therefore you are referred to corresponding literature (e. g. [2,4,9]). An excellent short introduction is [8].

The fuzzy shell FLOP is designed as a MDI application (*Multiple Document Interface*). After the start of the program an empty project is shown (see the following screenshot). The main window of the program is divided into three different areas:

- A toolbar directly below the window menu. It allows a quick access to the most important functions.
- The project tree in the left area. It contains all components of the current system structure, i. e. especially all lingustic variables with their fuzzy sets and the rule base. You can arbitrarily modify the width of the project tree with the mouse. If necessary it can completely be hidden via the  button.The project tree offers a powerful context menu

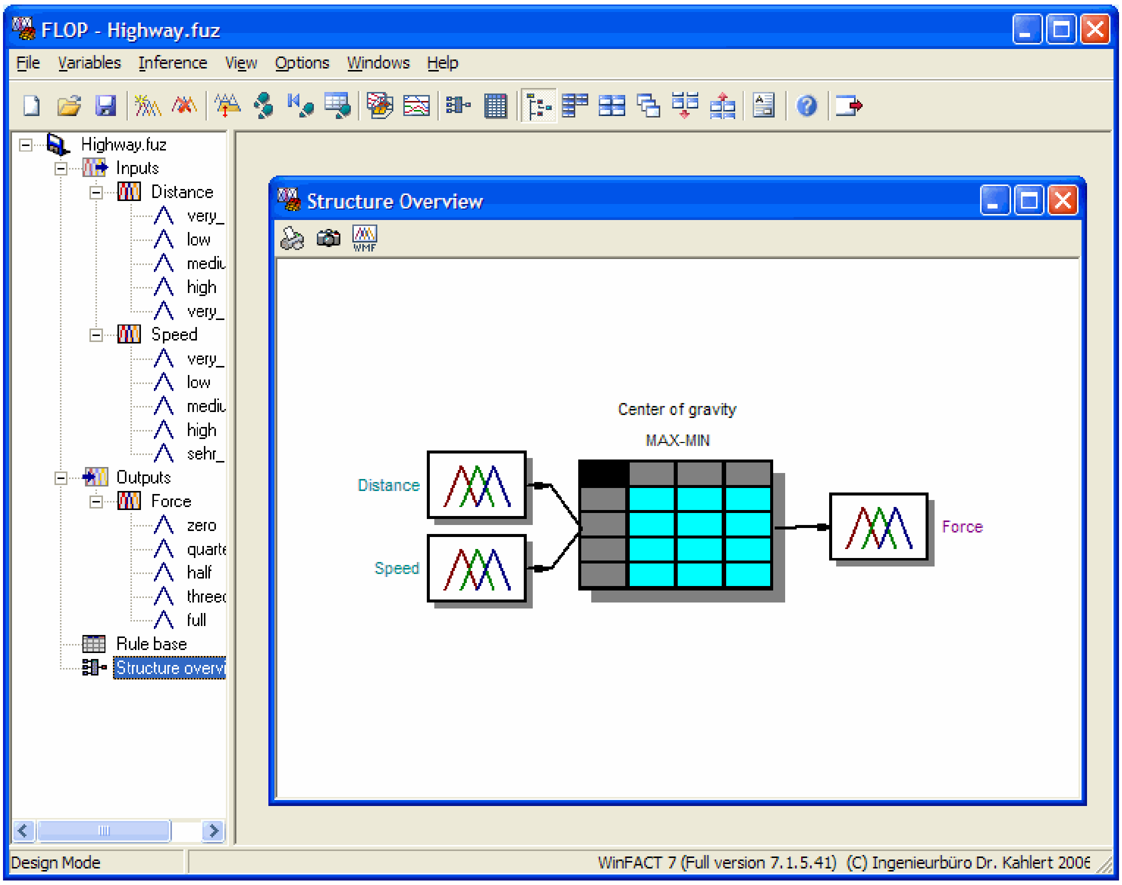
(popup menu) which appears after clicking with the right mouse button on an item in the project tree.

- The client area on the right of the project tree. After the program has been started it is empty at first. It will later contain all MDI child windows, e. g. the variable windows, the rule base etc.. These child windows can be moved to any position within the client area but not out of it.



Program window after the start (top) and after loading a project with three variable windows (bottom)

At any time the current system structure can be displayed in a separate window via the menu option VIEW | STRUCTURE OVERVIEW, the button or by clicking on the corresponding symbol in the project tree. Via the buttons of the window toolbar the window content can then be printed, copied to the clipboard or exported as WMF file.



The structure overview window

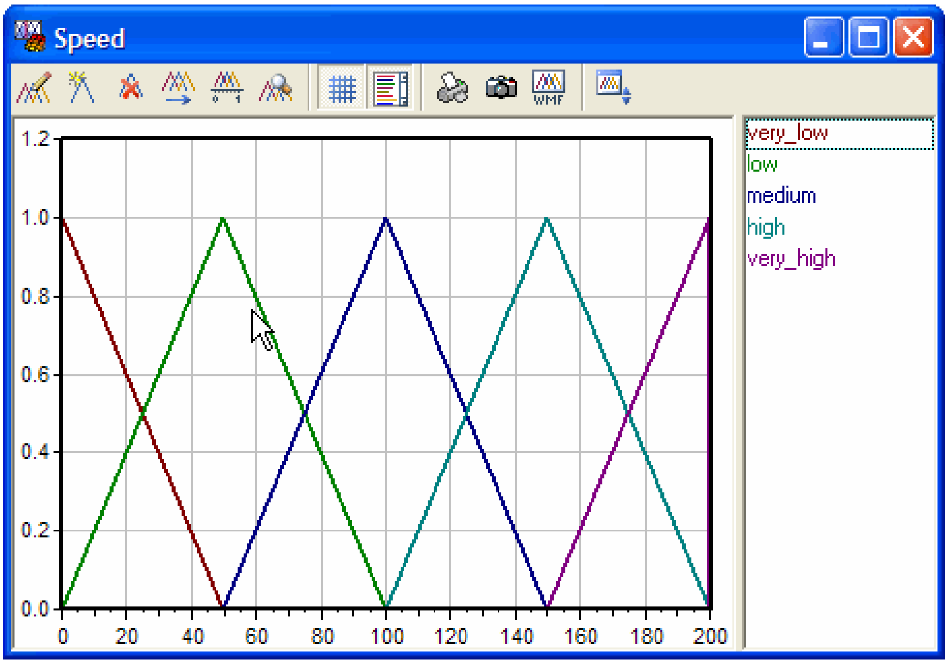
When saving a project the state (size, position etc.) of all windows (main and child windows) is saved together with the project data to a file with the extension FUZ. By that the program will be reloaded in exact the same state as it was when it was last saved.

A project can be operated in two different modes: the *design mode* and the *debug mode*. Not all functions of the design mode are available in the debug mode. Via the menu option INFERENCE | INTERACTIVE DEBUG MODE resp. the button you can switch from one mode to the other. The current mode is shown in the left area of the statusbar.

Linguistic variables and terms

Inserting new variables

The program is very clear because the user at any time gets an overview of the currently defined lingusistic variables, the corresponding linguistic terms and - if existing - the rule base. Each linguistic variable is represented in its own window which can be moved to any position and arbitrarily be minimized or maximized. Via the corresponding buttons of the window toolbar the content can at any time be printed, copied to the clipboard or exported in WMF format. You can arbitrarily modify the width of the term list at the right border of the window. Via the button the term list can also be hidden if necessary.



Variable window for a variable with six linguistic terms in the design mode

There are various ways to insert a new linguistic variable:

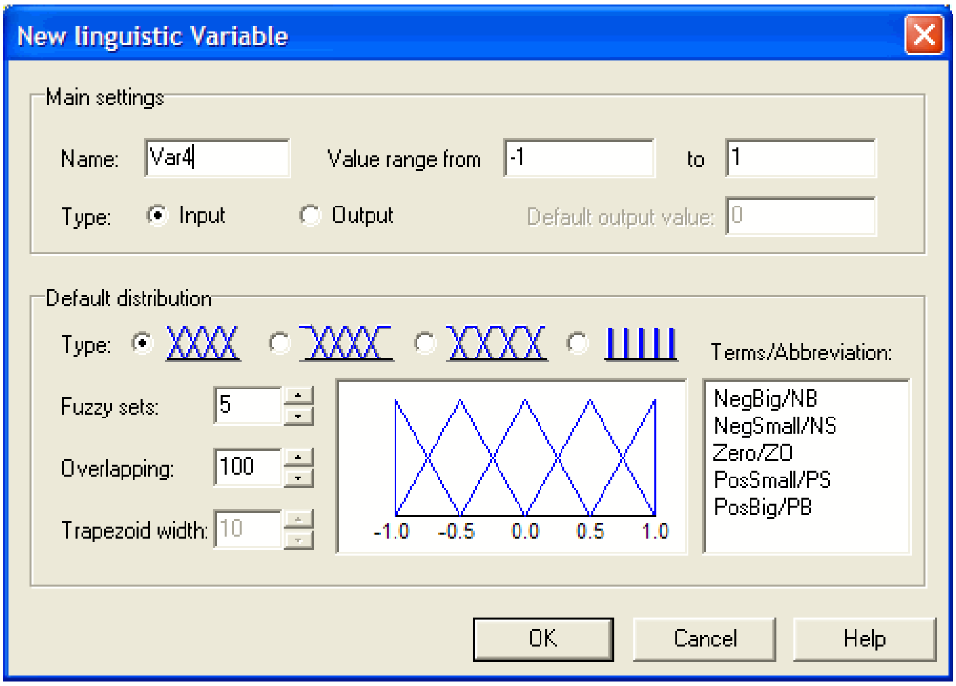
- The menu option VARIABLES | NEW LINGUISTIC VARIABLE...
- The button of the main window toolbar

- Clicking on the item input resp. output in the project tree and selecting the corresponding menu option in the appearing context menu.

A linguistic variable like e. g. *temperature* is defined by

- its *type* (input or output variable according to the premise resp. conclusion of the rules),
- its *name* (max. 15 characters),
- its *numerical range of values*,
- and a *default output value* for output variables in case that there is no active rule.

After the insertion of a new variable a dialog for the specification and presetting of the variables is displayed. The following screenshot shows the input dialog after having already entered the parameters for the variable *temperature*. A range of values from 0 to 50 was chosen with at first three fuzzy sets in shape of triangles with complete overlapping as presetting. The names of the fuzzy sets (linguistic terms) which are automatically set by the program and the corresponding abbreviations are listed in a list box on the right in the dialog window. They can directly be modified by clicking on them within this box. The abbreviation is only supported for keeping compatibility to older versions; it has no function in this latest version of FLOP.

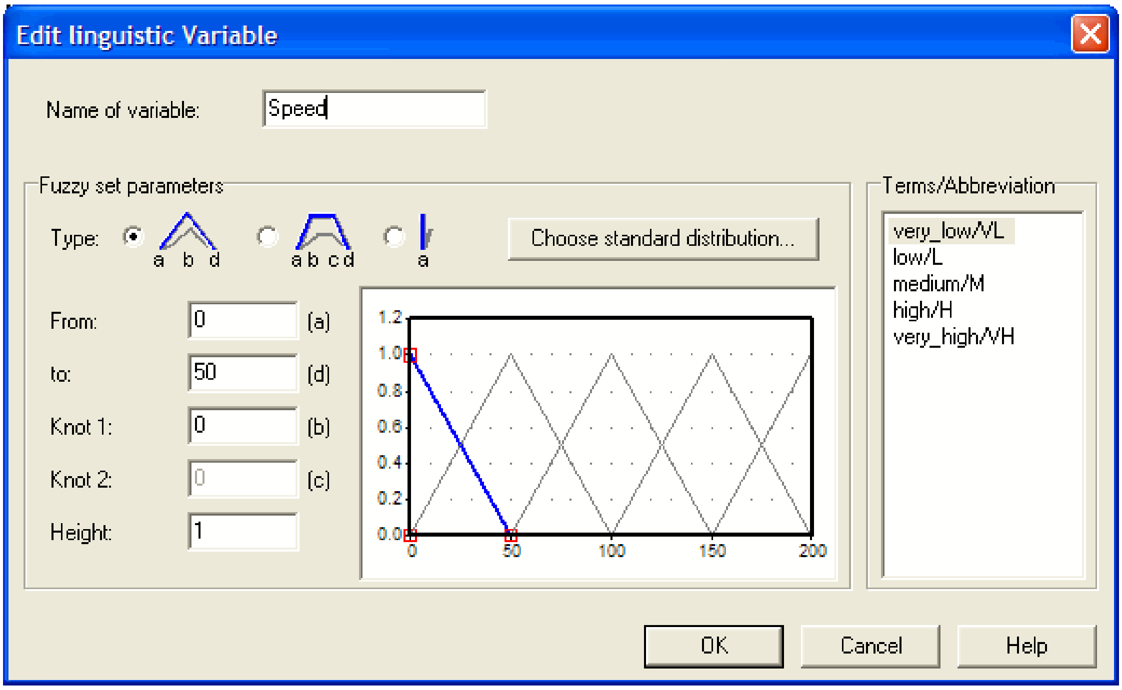


Dialog for the definition of new linguistic variables

Modification of variables

There are various possibilities for editing the fuzzy sets of an already existing variable or its renaming:

- Activating the variable window and clicking on the button resp. selecting the main menu option VARIABLES | EDIT LINGUSTIC VARIABLE...
- Doubleclick with the left mouse button within the variable window
- Clicking on the variable in the project tree (right mouse button) and selecting the option EDIT VARIABLE... in the context menu.

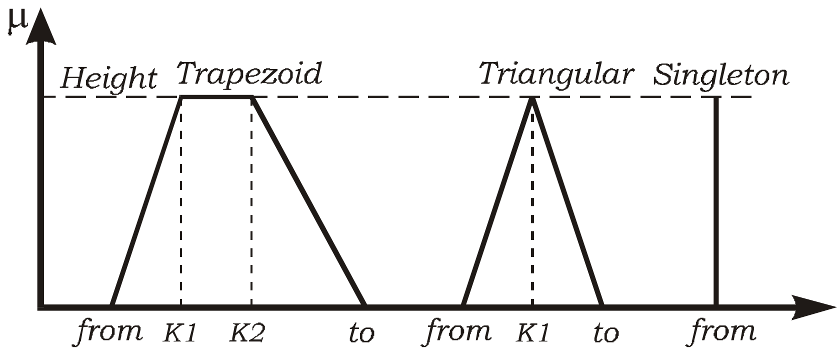


Dialog for the modification of fuzzy sets of a variable

The membership function which is to be edited is selected by clicking on the corresponding list item on the right in the dialog. All linguistic terms for the current variable are graphically displayed in the corresponding window. In addition the selected term is color highlighted and its characteristic knots marked. All changes are directly refreshed. By clicking on a term in the list it can be renamed.

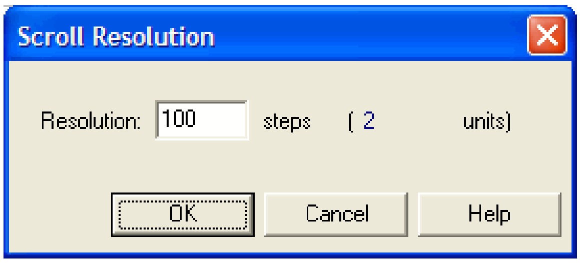
If *Singleton* was the selected type for the membership function you have only to enter a (crisp) value in the field *From*. The input field *Knot 2* is only selectable if the chosen fuzzy set type is *trapezoid*. By entering a *Height* value less than 1 subnormal fuzzy sets can be realized but they should normally not be used (subnormal sets are especially not allowed for the C-code-generation!).

The following screenshot shows all characteristic knots for all types of membership functions.



Characteristic parameters for each type of membership functions

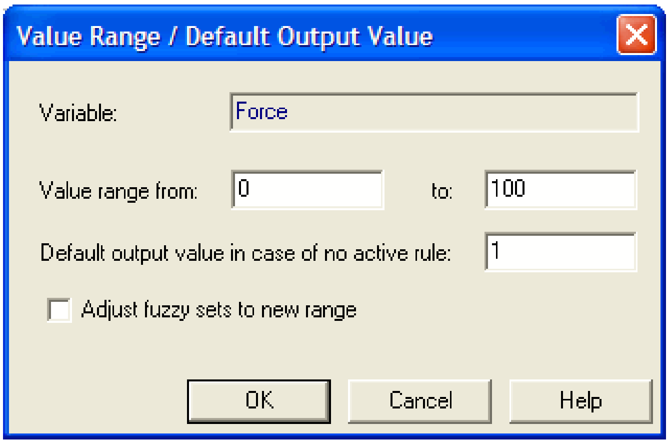
Instead of entering the numerical values directly the fuzzy sets can be edited graphically with the mouse. By clicking with the left button on a fuzzy set in the display window it at first gets activated. Afterwards each of the the characteristic knots which are marked by red squares can be moved with the left mouse button pressed. For the mouse controlled modification of a fuzzy set a default resolution of 100 base points (related to the range of values) is used. This value can be changed via the button of the variable window menu.



Dialog for setting the resolution of the variables

The button *Choose standard distribution...* is especially noteworthy. At any time it allows a quick standard setting of all membership functions of the selected linguistic variables according to the options when defining the variables for the first time.

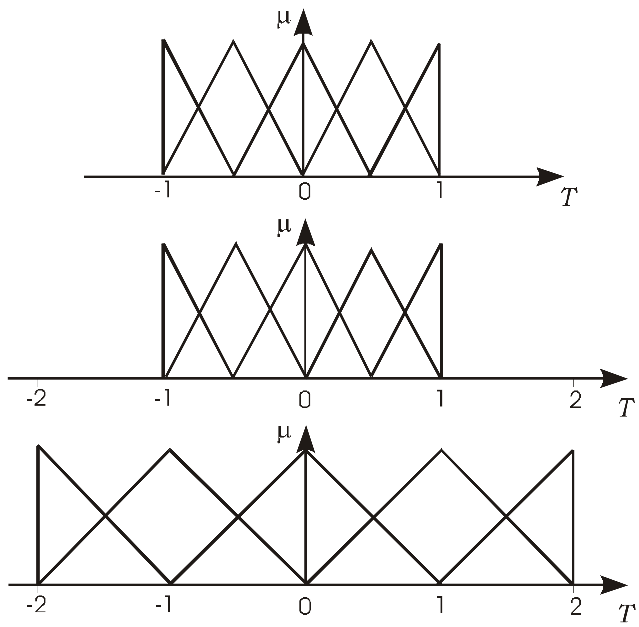
Also most of the other options for the modification of variables are accessible alternatively via the toolbar of the variable window, the main menu or the context menu of the project tree. So the range of values or the default output value (only output variables) can be modified via the button.



Dialog for the modification of the range of values or default output value of a variable.

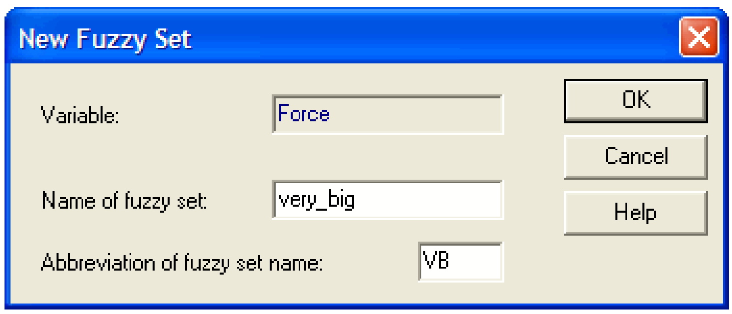
Hint: Is the fuzzy system later to be used under BORIS it is recommended to standardize in FLOP all in- and output variables to the range of values [-1, 1] resp. [0, 1] and to realize the adjustment to the actual ranges of values by connecting P-elements to the block in- resp. outputs in BORIS later. Doing it this way saves a lot of time.

When changing the range of values of a linguistic variable which already contains fuzzy sets you often wish to restandardize also the fuzzy sets themselves so that they can use the whole new range of values. This can be done via the option *Adjust fuzzy sets to new range*. If the option is not selected the linguistic terms remain at their defined positions. This option is very useful in case you realize later that the range of values of a linguistic variable may become smaller or has to be extended. By marking the checkbox *Adjust fuzzy sets to new range* the sets of this variable will be compressed resp. stretched corresponding to its newly defined range. The following graphic illustrates this option with an example.



Top: changing the range of values of a variable T from [-1, 1] to [-2, 2]
Middle: resulting constellation without adjustment of the fuzzy sets
Bottom: resulting distribution of the fuzzy sets with adjustment

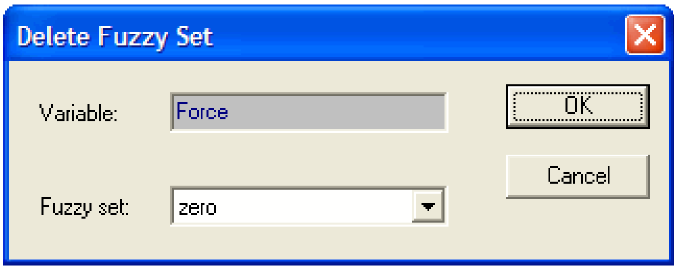
New terms for a variable can be defined by using the button.



Dialog for the definition of new linguistic terms

As mentioned the abbreviation is only still existing for keeping compatibility to older versions; leaving the field empty has no consequences. When defining the membership function it is at first initialized in such a way that it is symmetrical and tringular and the influence width is identical with the range of values of the corresponding linguistic variable. Therefore it has afterwards to be edited.

For deleting linguistic terms the button is used. The term which is to be deleted can be selected in the following dialog.



Dialog for deleting linguistic terms

When editing a variable afterwards the order of its terms may possibly be "mixed up". In this case the button can be used for resorting the terms.

Defining and editing a rule base

Representation modes of the rule base editor

For the creation and modification of a rule base there is a comfortable rule base editor, the *rule base window* as it is called. This window - if not currently visible - can be called in different ways:

- Via the main menu option VIEW | RULE BASE
- Via the button of the main menu toolbar
- By clicking on the rule base symbol in the project tree

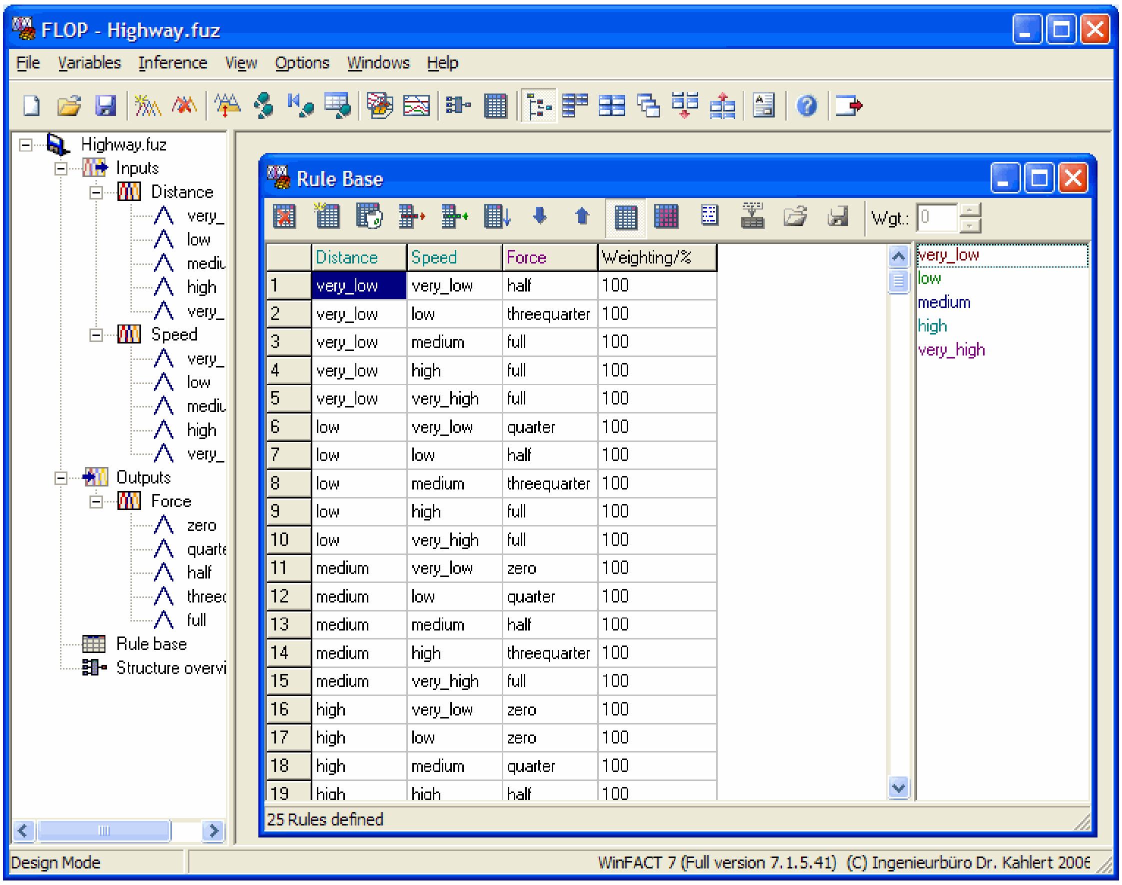
There are three different forms for representing the rule base within the window:

- Table form: Each column of the table contains a linguistic variable and each row a rule. This is the default mode for representing the rule base. If necessary it can be activated via the button in the toolbar of the rule base window.
- Matrix form: The linguistic terms of the first input variable are listed in horizontal and the terms of the second input variable are listed in verti-

cal direction. The inner cells contain the linguistic terms of the output variable. This representation mode is only available for fuzzy sets with two input variables and one output variable and can be activated via the button.

- Text form: A conventional text editor can be used for editing. This representation mode can be activated via the button.

The rule base window can arbitrarily be minimized, maximized or iconified. In contrast to the variable windows it can be closed if necessary; the rule base itself continue to exist.



Rule base editor for a system with two input variables and one output variable (here in table mode)

An existing rule base can completely be deleted via the button of the toolbar of the rulebase window after having confirmed to carry out this action.

The table mode of the rule base editor

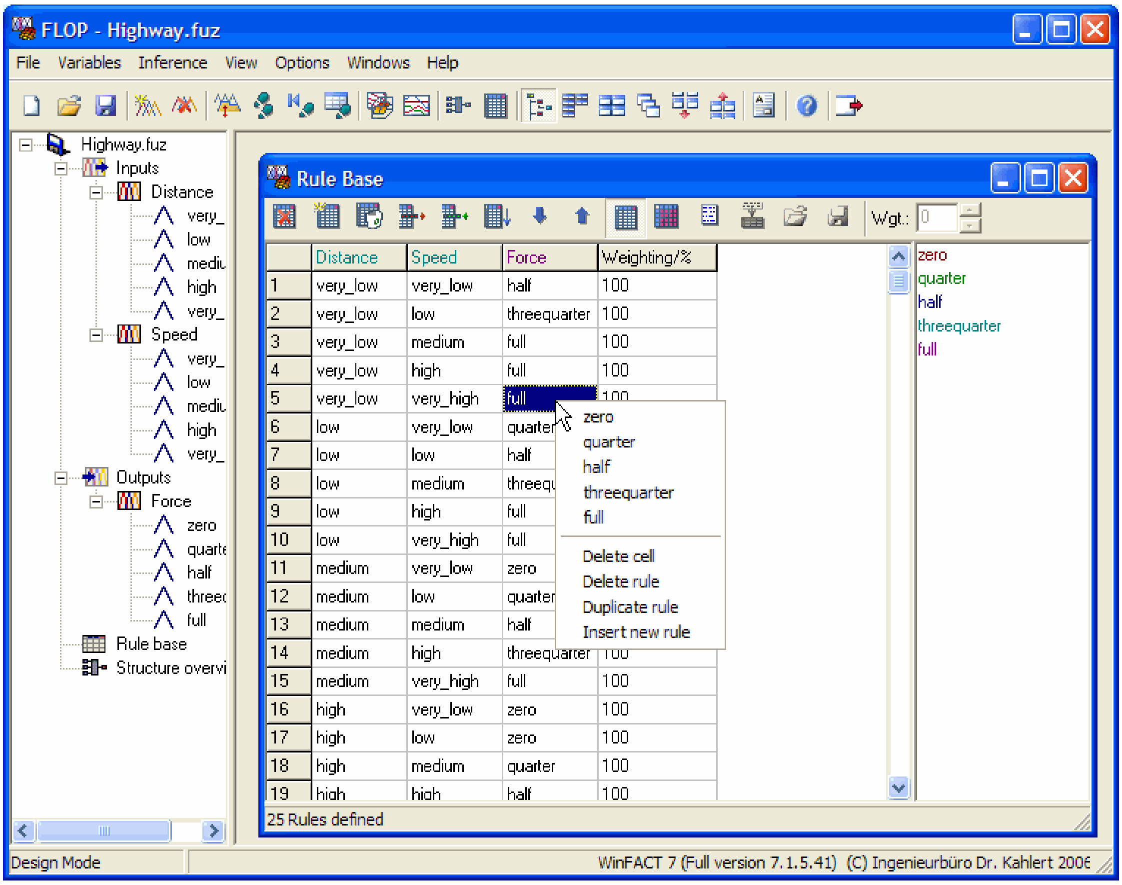
In the table mode of the rule base editor each column of the rule matrix corresponds to one linguistic variable; input variables are labelled turquoise, output variables are labelled purple colored. Each row of the matrix corresponds to one rule. In addition the last column of the matrix contains the weighting (extent of confidence) of the rule as a percentage from 0 ("no confidence") to 100 ("full confidence"); each new rule has at first a weighting of 100% as default.

Inserting and deleting linguistic terms

Most options of the rule base editor are on the one hand available via its toolbar but on the other hand also via a context menu which is displayed when clicking on a cell with the right mouse button. E. g. is a cell to be filled with a linguistic term there are two possible ways:

- Clicking on a cell with the *left* mouse button. The list at the right border of the window then shows all terms of the corresponding variable. By doubleclicking on an entry it is inserted in the selected cell.
- Clicking on the cell with the *right* mouse button. In the following context menu the term which is to be inserted is selected (see the following screenshot).

Both ways can be used for filling several cells - one below the other - of one column all at once with the same term. When selecting the left mouse button has to be kept pressed.



Context menu of the rule base editor in table mode

The context menu allows the negating of terms of input variables (NOT-operator). Within the cell such a negated partial premise is marked by a tilde as prefix (e. g. *~small*).

Single or several cells - one below the other - of one column can be deleted by selecting them as described above and choosing the option DELETE CELL in the context menu.

Editing complete rules

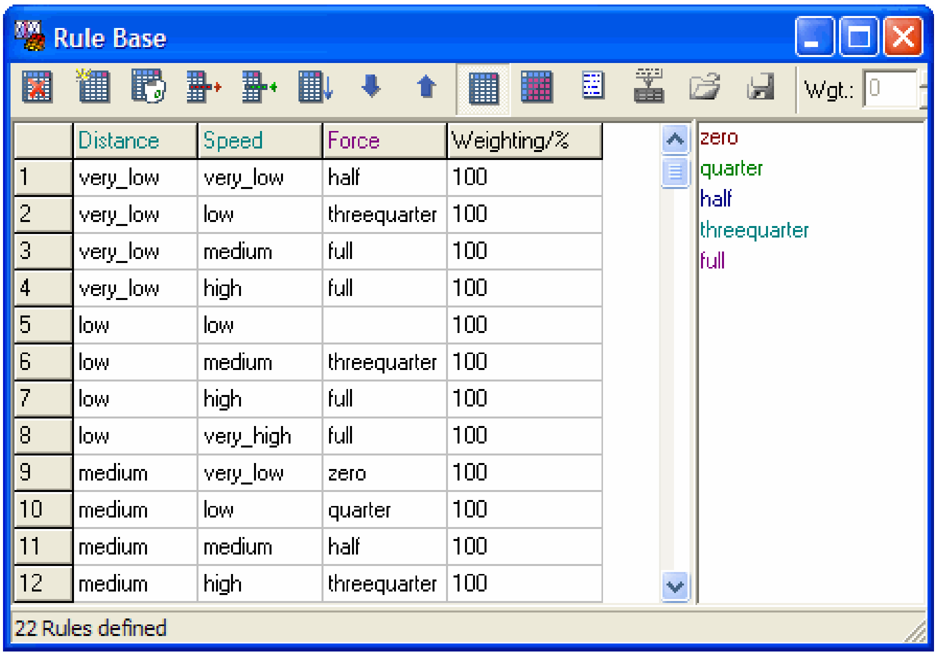
Besides the modification of single cells of the rule base the editor offers numerous functions for editing complete rules (table rows of the rule table). To select one or several rules *any* cell or cells side by side within one row can be selected. It is not necessary to select the cells all at once.

After having selected one or several rules - one below the other - the following operations are available.

- *Deleting* all selected rules via the context menu or the  button.

- *Moving* all selected rules up or down via the resp. button. Alternatively single rules can be moved with the mouse by clicking on the rules in the left, gray column of the table (rule number) and moving them to their new position with the left mouse button kept pressed.
- *Duplicating* a rule via the context menu (only allowed for single rules). The duplicated rule is inserted directly below the selected rule.
- *Inserting* a new ("empty") rule above the selected one via the button.

The button enables the removing of invalid rules from the rule base. Rules are invalid if there is no entry for at least one output variable (see the following screenshot). This clear-up procedure is automatically executed when closing the rule base window. In additon all rules can be sorted by linguistic terms in the corresponding premises via the button.

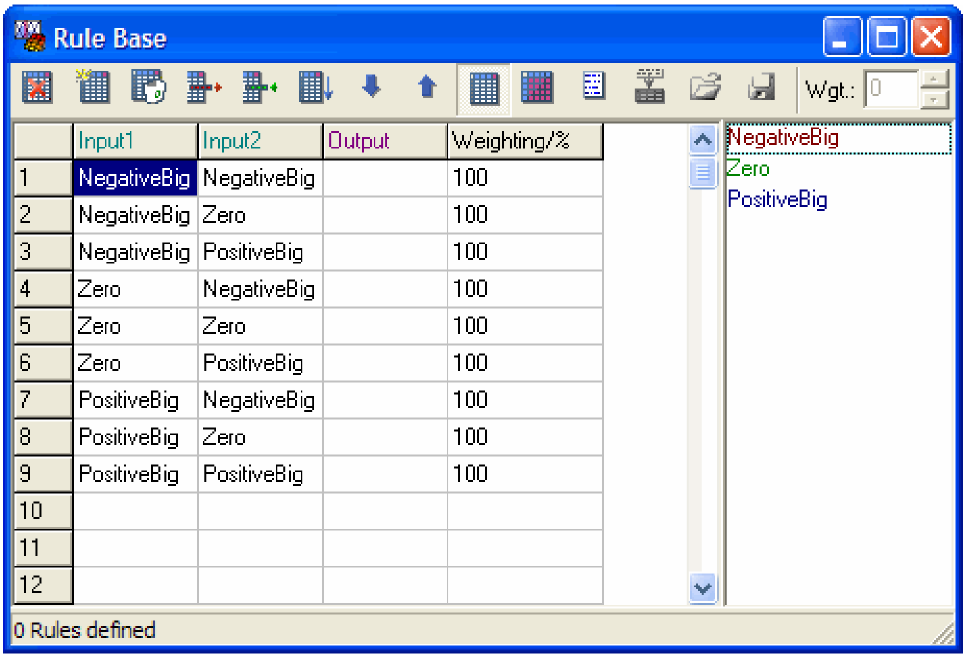


	Distance	Speed	Force	Weighting/%
1	very_low	very_low	half	100
2	very_low	low	threequarter	100
3	very_low	medium	full	100
4	very_low	high	full	100
5	low	low		100
6	low	medium	threequarter	100
7	low	high	full	100
8	low	very_high	full	100
9	medium	very_low	zero	100
10	medium	low	quarter	100
11	medium	medium	half	100
12	medium	high	threequarter	100

Example for an invalid rule: rule 5 contains no conclusion and is therefore currently invalid

Automatic generation of premises

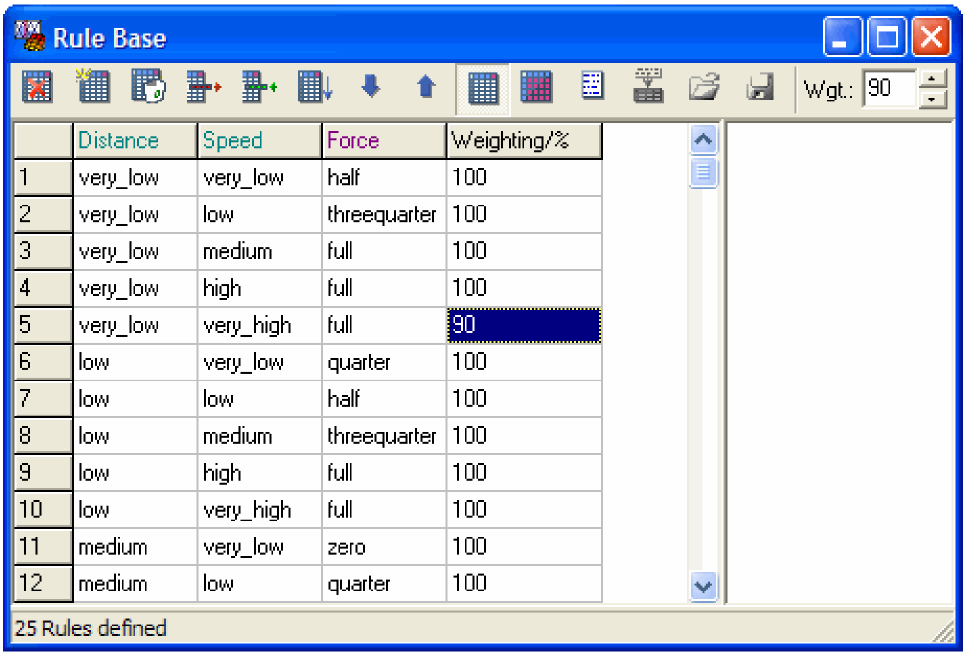
For many fuzzy sets a *complete* rule base is desired to the effect that for any possible combination of input variable terms (partial premises) one rule does exist. In the table mode the rule base editor allows the automatic generation of the corresponding IF-part of the rule base so that only the conclusion terms have to be entered by the user. That's what the button is for. The following screenshot shows the "rule framework" for a fuzzy set generated in this way with two input variables, each of them has three lingusitic terms. That is why altogether nine rule frames are generated.



Example for the automatic generation of premises

Changing the rule weighting

Each rule has a rule weighting of its own with a default value of 100% (full weighting). Is the weighting of a rule to be changed select at first the corresponding field of the rule table. The desired weighting can then be set in 1%-steps in the input field *weighting* of the toolbar. By selecting several cells the weighting can be changed simultaneously.



Changing the weighting of a rule

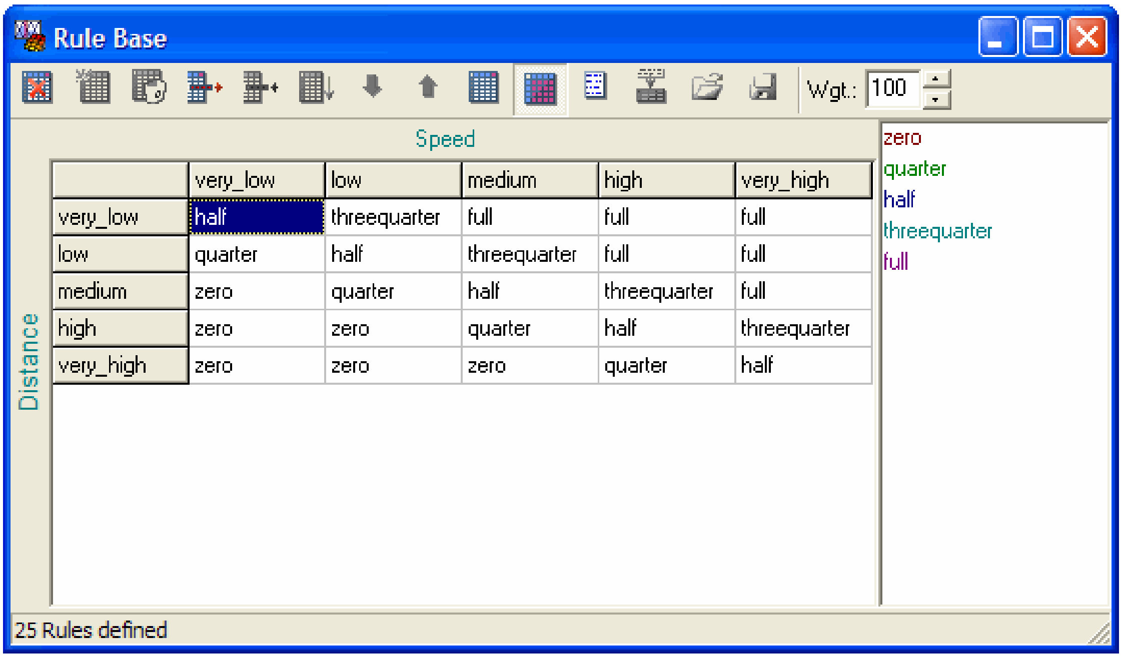
The matrix mode of the rule base editor

For a specially clear representation of fuzzy systems with two input variables the matrix mode of the rule base editor is predestined. It is available on conditions that:

- The fuzzy system has two input variables and one output variable.
- All already existing rules are complete regarding their premise (i. e. no partial premise is allowed to be empty).
- No partial premise is negated.

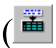
In the matrix mode the premises don't have to be entered because they are automatically determined due to their matrix structure. The user has only to enter suitable conclusion terms into the inner matrix cells. This form of representation allows to check the completeness of the rule base directly at a glance.

The weighting of rules can be modified in the matrix mode too by clicking on the corresponding cell(s).

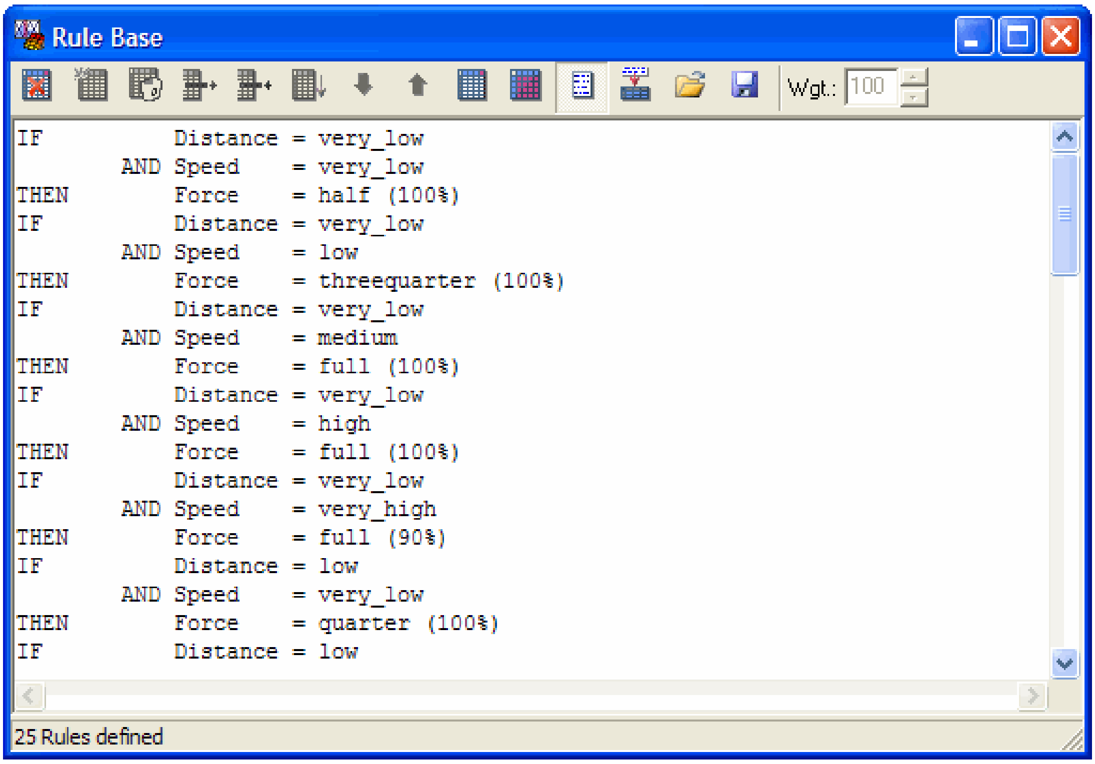


Rule base in the matrix mode (here with sample file HIGHWAY.FUZ)

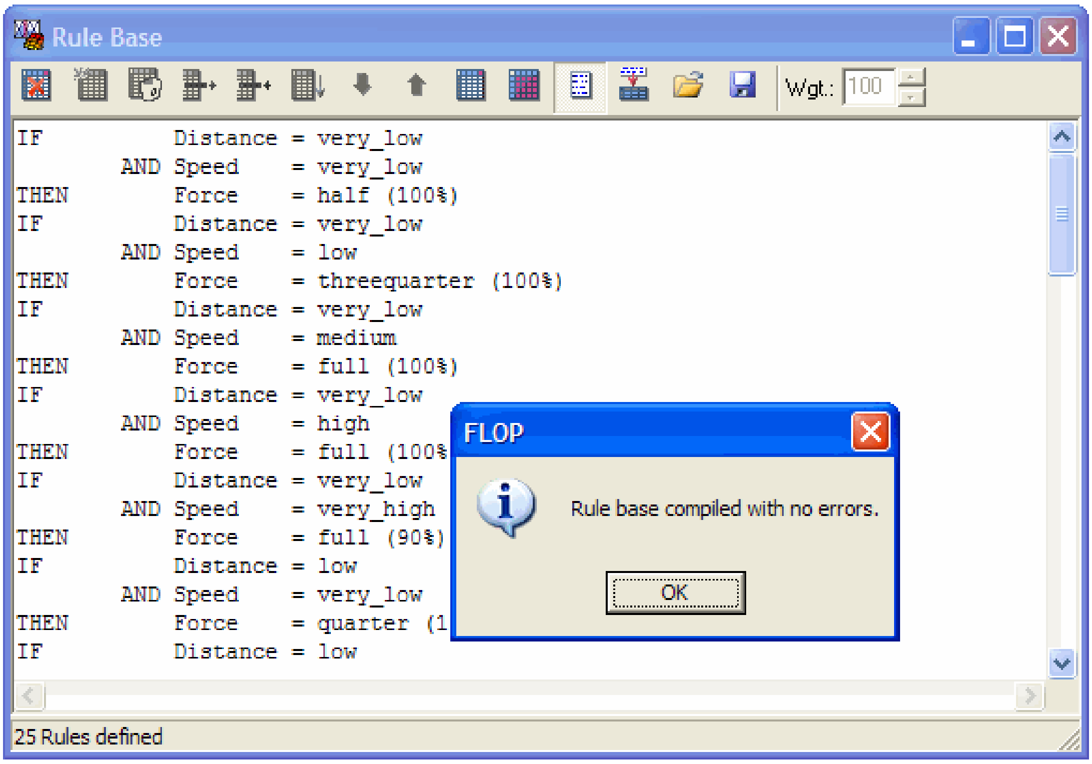
The text mode of the rule base editor

Like the table mode the text mode is available for any system and rule structures. In this form of representation the editor is equivalent to a simple text editor where the rules can be entered as "plain text". Before switching back to one of the other modes (table resp. matrix mode) the rule base has at first to be recompiled ( button). In the course of this it is automatically checked for syntax errors.

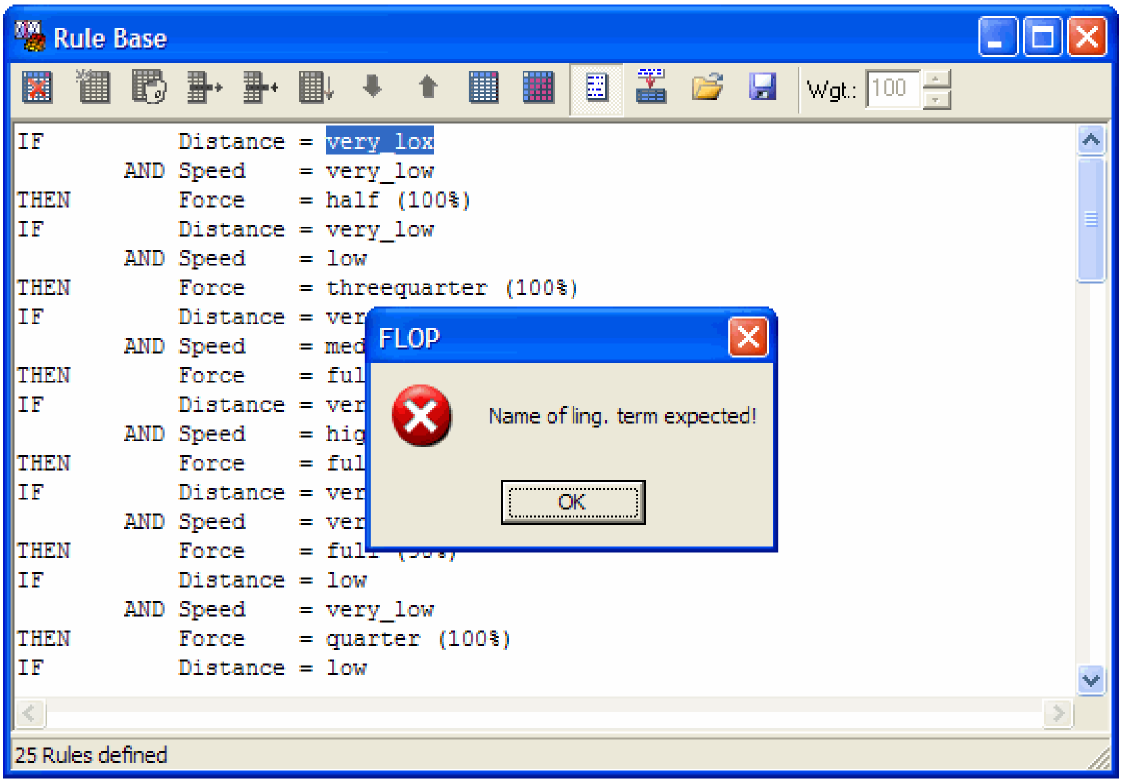
The names of linguistic variables and terms can be abbreviated by a point if this abbreviation is unambiguous (e. g. the variable *Speed* by *Sp.* in the sample file HIGHWAY.FUZ). Ambiguous abbreviations as well as other syntax errors are pointed out to the user when compiling the rule base. If an error is detected during compilation the corresponding part in the edit window is selected and a corresponding message is displayed.



Rule base editor in the text mode (sample file HIGHWAY.FUZ)



Names of variables and terms may be abbreviated.
The rule base compiler automatically recognizes this and compiles the rules correctly.



Here the compiler has detected a typing error in the first rule and has marked the corresponding part.

The following syntax diagrams illustrate the structure of the rules in the text mode. Relational operators like e. g. = have to be enclosed by blanks. For negating partial premises the operators

= *NOT*

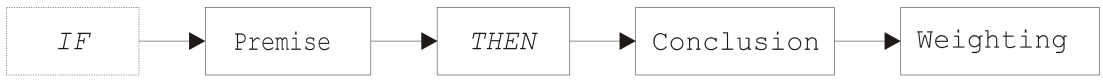
= ~

!=

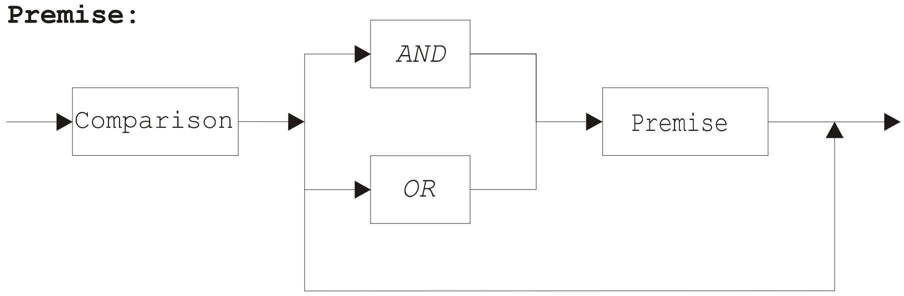
are alternatively allowed.

Via the and buttons the rule bases can also be read from text files or saved to text files (file extension *RB*). So the rules can be e. g. created and modified in other word processing programs and then be transferred to the rule base editor.

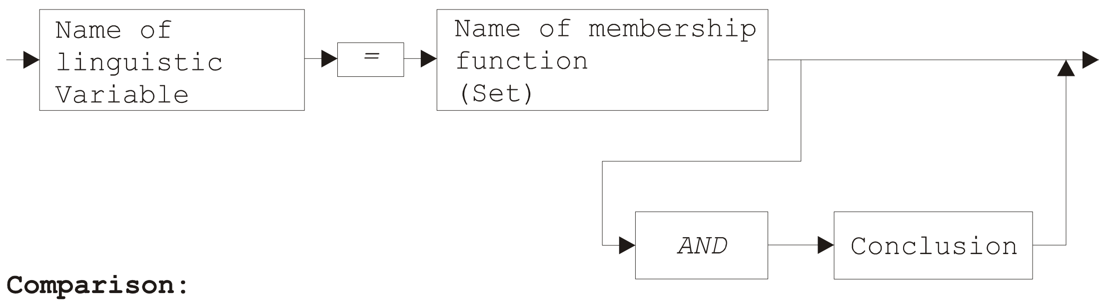
Rule:



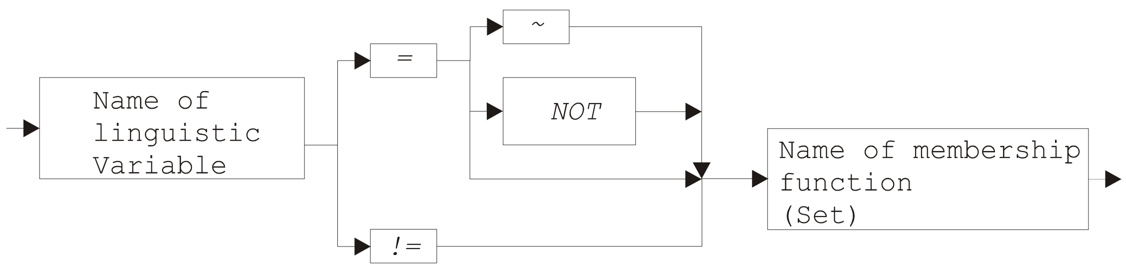
Premise:



Conclusion:



Comparison:



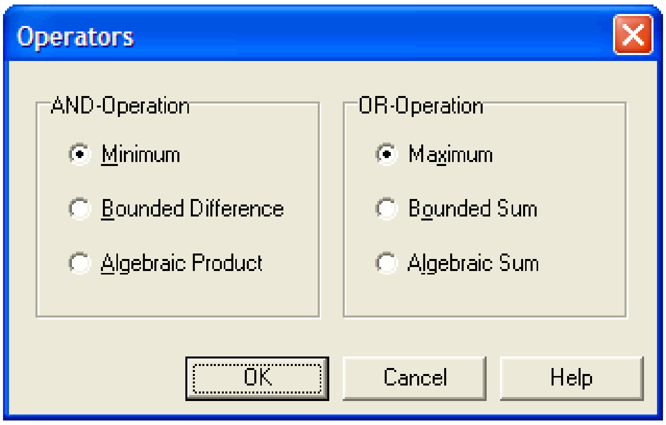
Weighting:

Syntax diagram of the text editor for rules

Operators, inference mechanism and defuzzification

Operators for AND- and OR-operations

FLOP offers various operators for the AND- and OR-operation of the partial premises. Via the menu option INFERENCE | OPERATORS... the desired operators can be selected (see the following screenshot).



Dialog for the selection of the operation operators

As default the MIN operator is used for the AND- and the MAX operator is used for the OR-operation. Alternatively there are the following other operators to choose from:

For the AND operation:

- Bounded-Difference-Operator

$$(\mu_1 \mathbin{\hat{-}} \mu_2)(x) := \mathrm{MAX}(0, \mu_1(x) + \mu_2(x) - 1),$$

- Algebraic-Product-Operator

$$(\mu_1 \mu_2)(x) := \mu_1(x) \cdot \mu_2(x).$$

For the OR-operation:

- Bounded-Sum-Operator

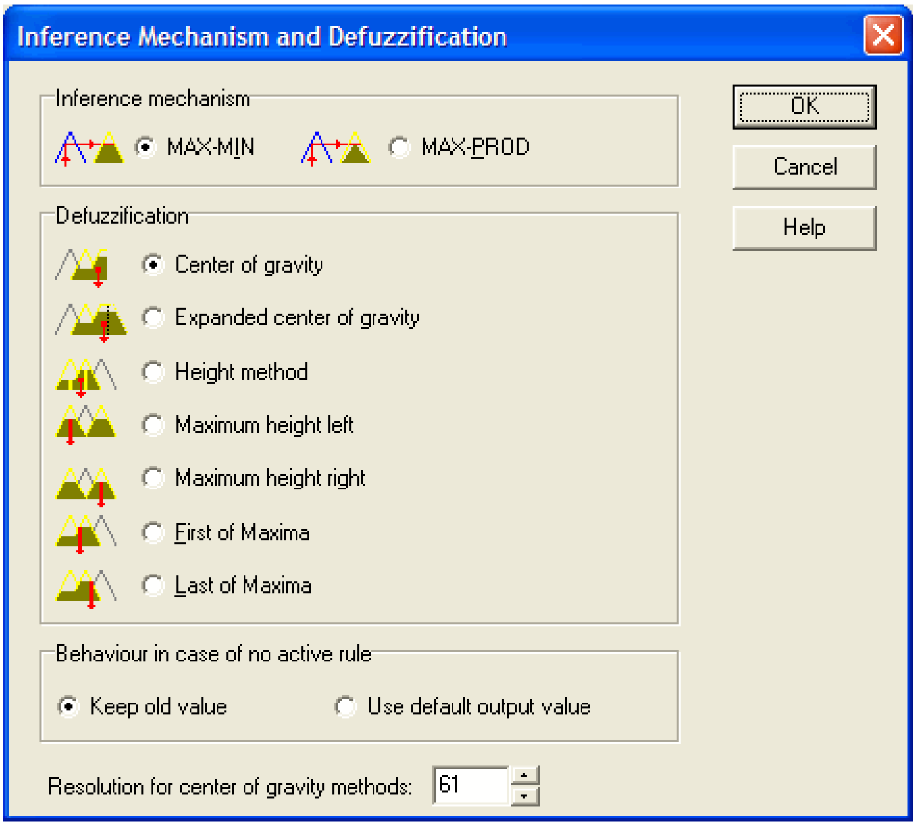
$$(\mu_1 \hat{+} \mu_2)(x) := \mathrm{MIN}(1, \mu_1(x) + \mu_2(x)),$$

- Algebraic-Sum-Operator

$$(\mu_1 \oplus \mu_2)(x) := \mu_1(x) + \mu_2(x) - \mu_1(x) \cdot \mu_2(x).$$

Inference mechanism and defuzzification

Via the menu option INFERENCE | INFERENCE MECHANISM AND DEFUZZIFICATION... inference mechanism and defuzzification can be selected.



Dialog for selecting inference mechanism and defuzzification

In the corresponding dialog you can select either MAX-MIN inference or MAX-PROD inference. Furthermore there are various defuzzification methods to choose from via the checkbox group *defuzzification*. They are explained in detail in [2,4]:

- Center of Gravity - Method

In the (original) center of gravitiy method the abscissa value of the centroid S of the resulting output fuzzy set $\mu_{res}(y)$ is selected for the crisp output value y_{res}. The accurate formula is

$$y_{res} = \frac{\int\limits_0^\infty y\mu_{res}(y)\mathrm{d}y}{\int\limits_0^\infty \mu_{res}(y)\mathrm{d}y}.$$

The integral is evaluated by program internal numerical integration. Its resolution can be set by the user.

- Modified center of gravity method (with extended border)

  This method is in principle equivalent to the original center of gravity method but both border sets of the output variable are symmetrically extended so that the minimum resp. maximum possible crisp output just corresponds with the range of values of the output variable. The result only differs from that of the original center of gravity method if one of the border sets influences the resulting output fuzzy set.

- Approximate center of gravity method (height method, center of gravity method for singletons)

  The original center of gravity method is very computing-time consuming. Therefore it is generally recommended to use an approximation formula for the center of gravity. This is

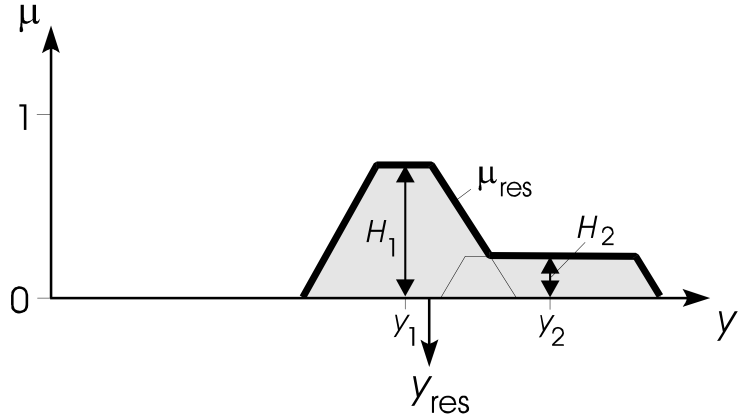
$$y_{res} \approx \frac{\sum\limits_{i=1}^{m} y_i H_i}{\sum\limits_{i=1}^{m} H_i}$$

with

m:	the number of rules
H_i:	the match of degree of i-th rule
y_i:	the abscissa value of the center of gravity of the i-th output set

For the fuzzy set shown below we get for example the relation

$$y_{res} \approx \frac{y_1 H_1 + y_2 H_2}{H_1 + H_2}.$$



Approximation formula for the center of gravity method

If singletons are defined for the output variable *y* the approximation formula corresponds with the exact equation.

- Defuzzification according to maximum height (maximum method)

  Only the rule with the maximum match of degree is used. The modal value of the corresponding result fuzzy set provides the crisp output variable. If there are several rules with the maximum match of degree the output fuzzy set at the extreme left or right can be selected. Therefore it is called the maximum left resp. the maximum right method.

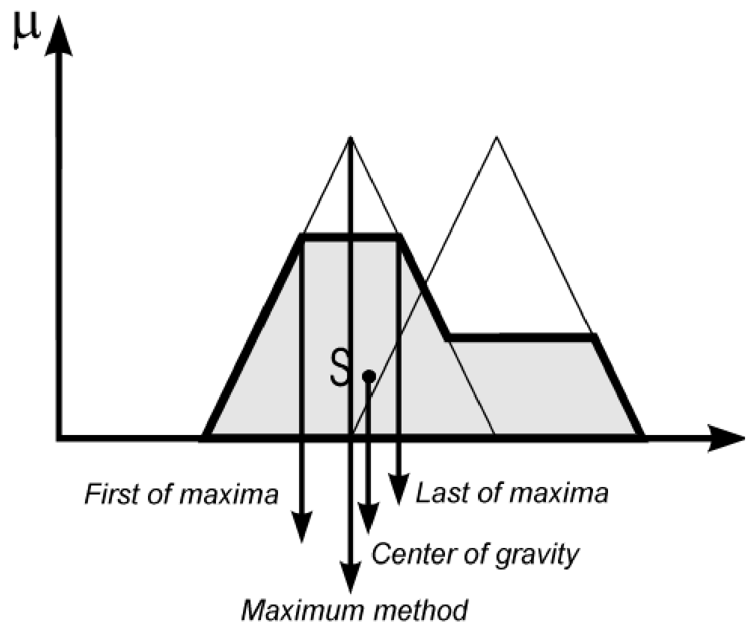
- First of Maxima-Method

  Only the rule with the maximum match of degree is used and the corresponding result fuzzy set is cut off at the level of the match of degree. The left knot of the resulting fuzzy set provides the crisp output variable [4].

- Last of Maxima-Method

  Only the rule with the maximum match of degree is used and the corresponding result fuzzy set is cut off at the level of the match of degree. The right knot of the resulting fuzzy set provides the crisp output variable [4].

The following graphic illustrates the different defuzzification method with an example.



Methods of defuzzification

In addition via the dialog the behaviour of the system can be specified in case that none of the defined rules is active. There in principle are two possible ways:

- The last valid output value is kept.
- The output is a fixed, output variable-dependent default value which can be set at the specification of the variable.

The parameter *Resolution for center of gravitiy methods* specifies the resolution used for the approximation of the integral in the center of gravity methods. The default value is 61.

System analysis in the debug mode

After the specification of all linguistic in- and output variables, the rule base, the operation operators as well as the inference mechanism and the defuzzification method the fuzzy system can be tested for its correct functioning. For that various tools for analysis are available which are especially advantageous in the interactive debug mode of FLOP.

All following considerations are based on the sample file HIGHWAY.FUZ

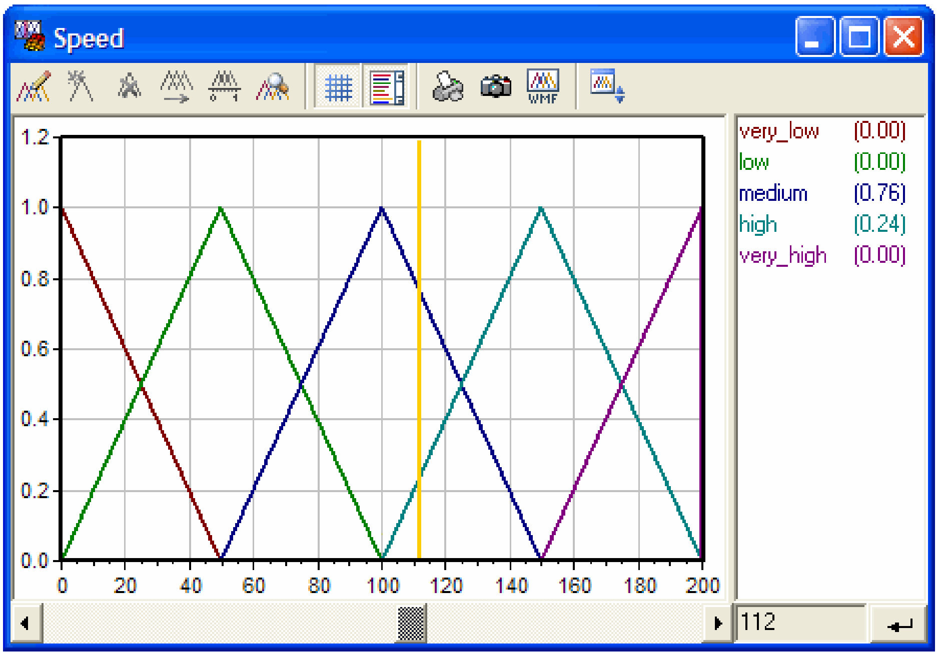
Activating the interactive debug mode

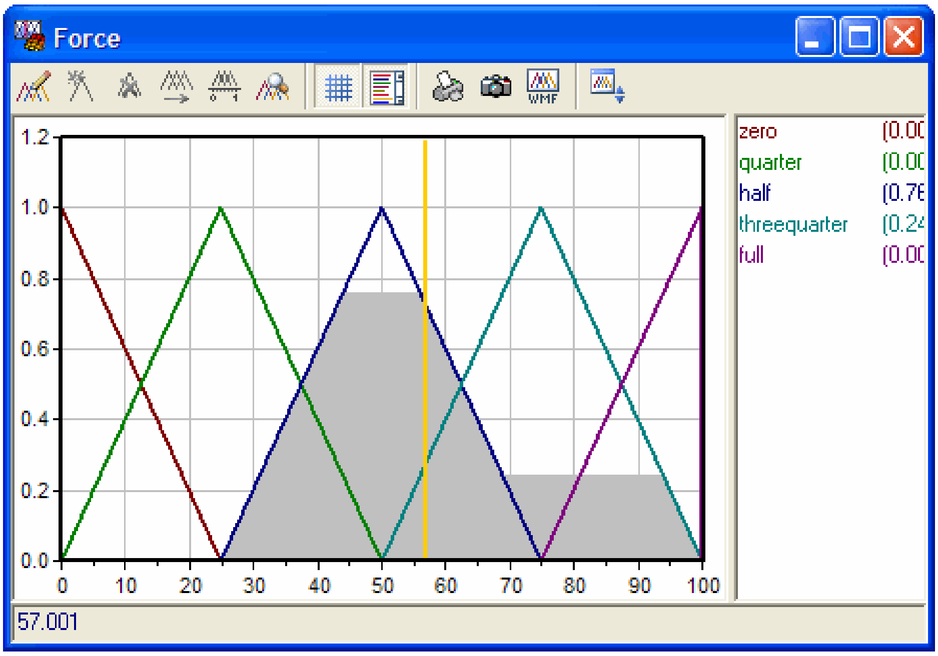
For the system analysis the program has to be in the interactive debug mode which can be selected via the menu option INFERENCE | INTERACTIVE DEBUG MODE resp. the  button. The interactive debug mode is confirmed in the statusbar of the program by a corresponding message.

As long as FLOP is in the debug mode the structure of the fuzzy system cannot be modified (e. g. inserting or deleting of linguistig variables). It is however possible to modify e. g. single fuzzy sets and change the rule base (inserting and deleting rules or changing rules).

Variable window in the debug mode

The variable windows of the in- and output variables look differently after the debug mode has been activated (see the following screenshots).



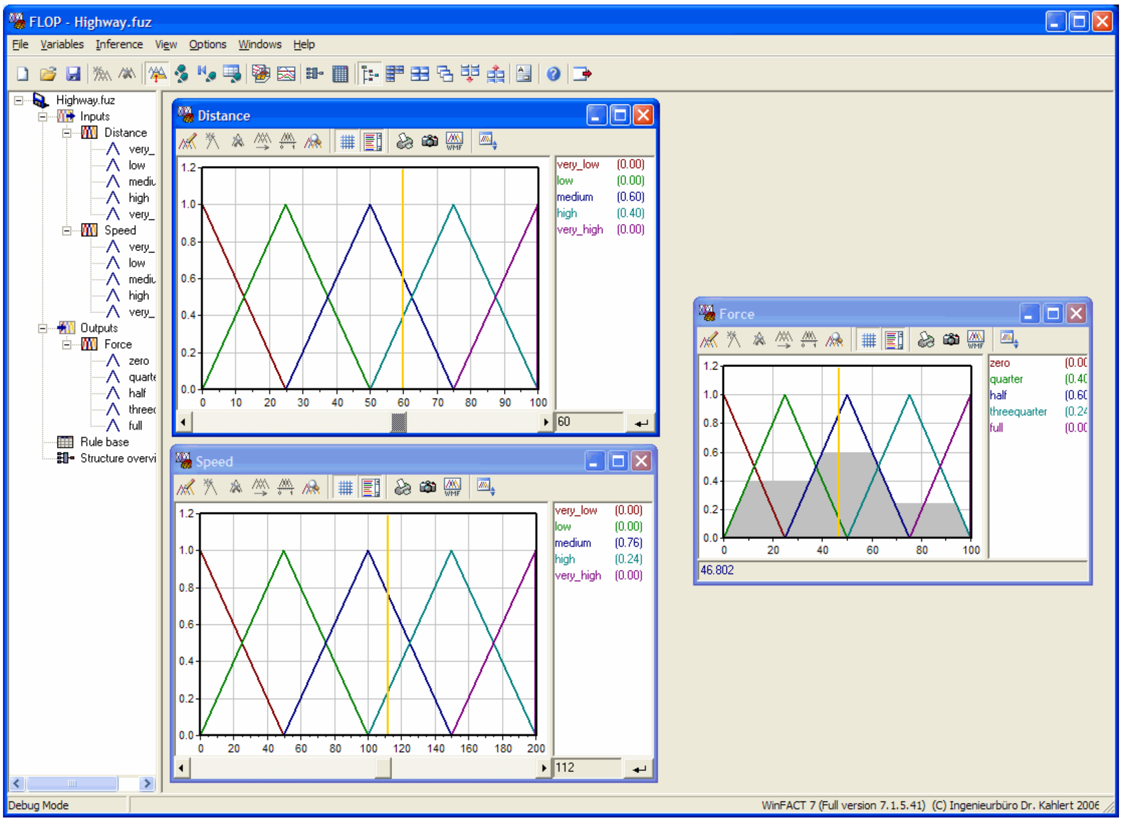


Variable windows for input variables (top) and output variables (bottom) in the interactive debug mode

Windows of input variables get a scrollbar at the lower border of the window as well as an edit field with button below the term list. Via the scrollbar or the edit field (and a click on the button on the right of the edit field) the current value of the input variable can be set. The default resolution for the scrolling process is 1% of the range of values of the variable. This value can be changed via the button. In addition the current input value is represented by a bar in the paint area of the window. In the term list of the window the evaluated degree of membership for the current input value is displayed beside every term.

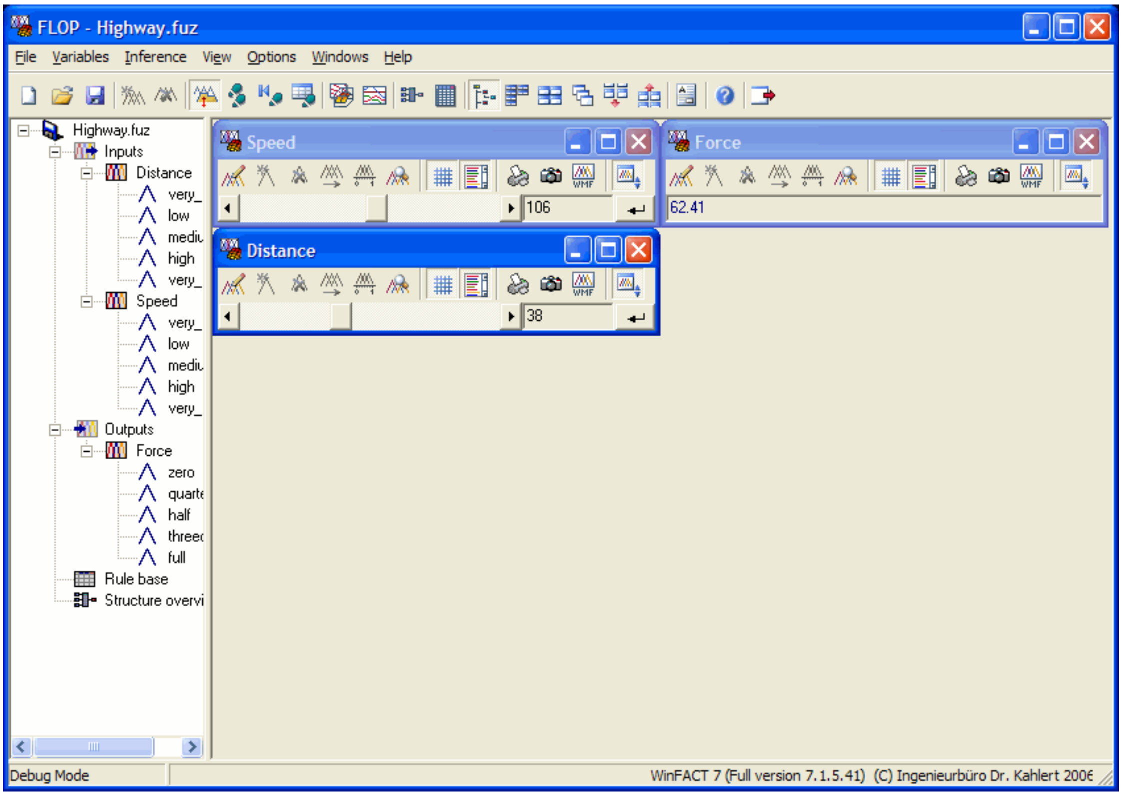
Windows of output variables have a statusbar at their lower border which shows the current crisp output value of the variable. At every modification of an input variable of the fuzzy system all output variables are automatically recalculated and the corresponding variable windows are refreshed. In addition the crisp output value is represented by a bar in the paint area of the window. Besides the resulting fuzzy set is color marked. In the term list, finally, the current matches of degree for all terms of the output variable are listed.

That way the behaviour of the fuzzy set can be analyzed step by step by modifying single input values and observing the effect of this modification on the output values (single step analysis).



Single step analysis with sample file HIGHWAY.FUZ

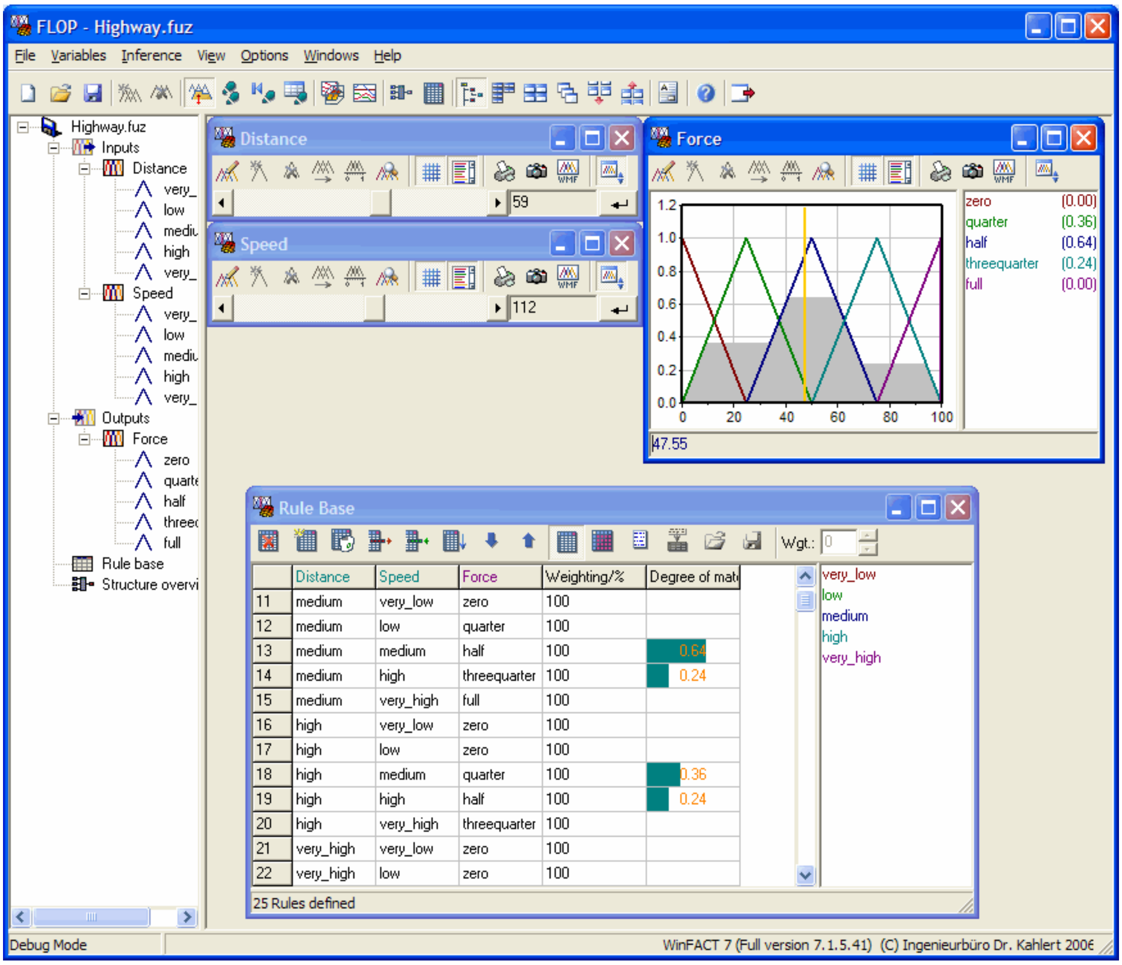
If only the numerical values of in- and output variables are of interest or the system contains many variables it might be useful to minimize some or all variable windows in such a way that only the relevant information is represented. For that the client area of the variable window can be hidden via the  button so that only the scrollbar (input variables) resp. the statusbar (output variables) is visible. By selecting the main menu option WINDOWS | ARRANGE ALL VARIABLE WINDOWS IN SHUT MODE or via the button of the main window it is possible to minimize and automatically arrange the variable windows all at once (Hint: For a correct display this option should be selected only *after* having switched to the debug mode) .



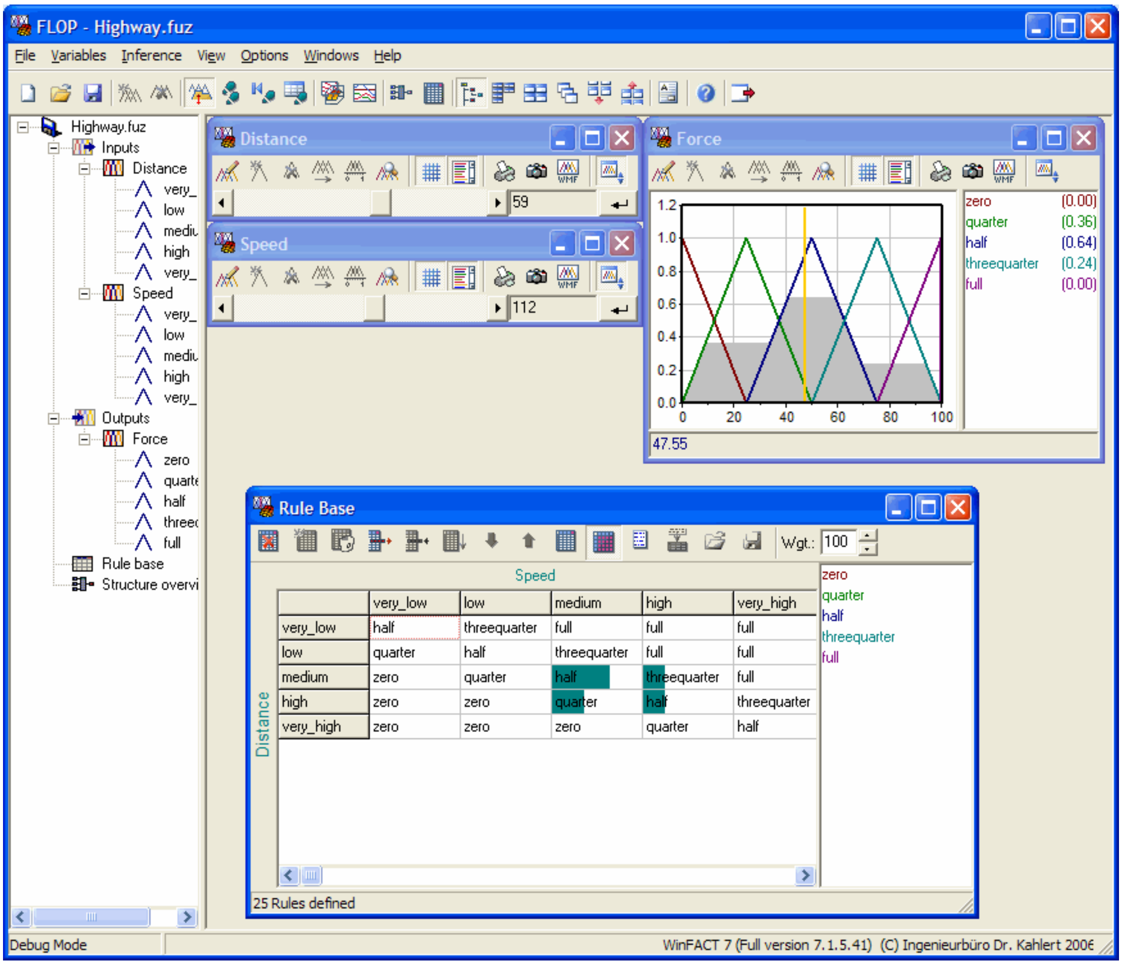
Automatic positioning of all shut variable windows:
input variables are on the left one below the other, output variables on the right

The rule base editor in the debug mode

The rule base editor too looks differently in the debug mode because for each rule the current match of degree is shown. In the table mode an additional column is displayed, in the matrix mode the output is shown directly in the inner area of the matrix (conclusion part). On each inference step (i. e. change of the value of the input variable) the display is automatically refreshed.



Rule base editor in the table mode with activated debug mode:
as you can see the rules 13, 14, 18 and 19 of the current input values are active.



The same situation but the rule base editor is in matrix mode: here the match of degree is represented only graphically within the corresponding cell of the matrix.

All options of the rule base editor are also available in the debug mode. This means that on the one hand existing rules can be modified, on the other hand rules can also be deleted or inserted.

Representation of characteristic curves and maps

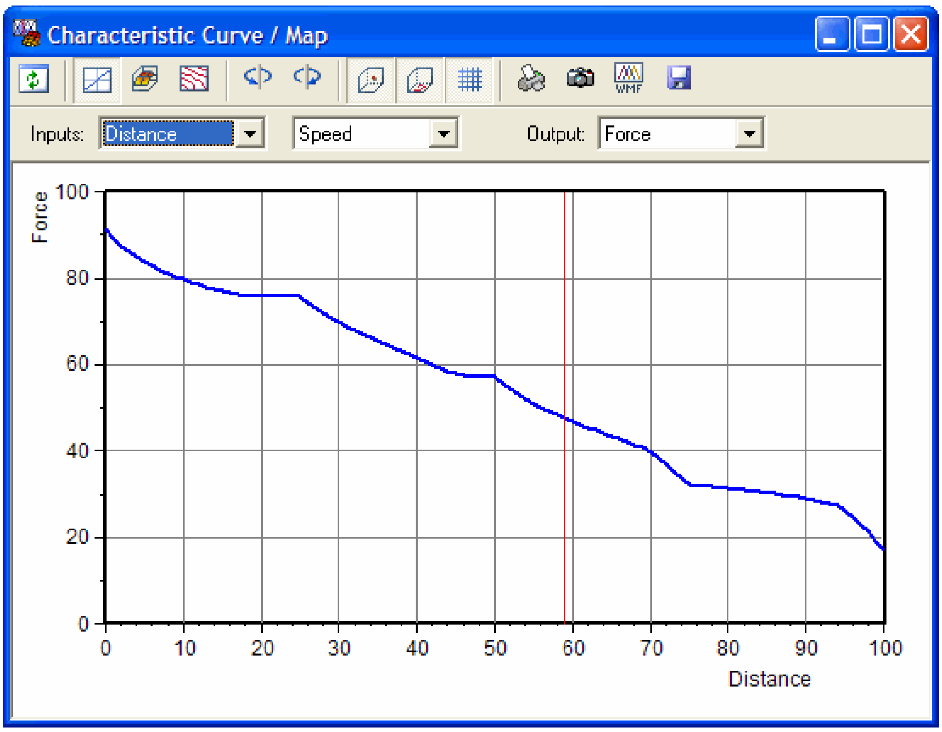
To analyze the behaviour of the fuzzy system globally, i. e. over the whole operating area, it can be represented in the form of characteristic curves or maps resp. contour lines. The corresponding analysis window can at any time be opened via the menu option VIEW | CHARACTERISTIC CURVE/MAP or the button. It can arbitrarily be minimized or maximized, iconified or again closed. Like all other windows it is automatically refreshed in the debug mode on every change of input values, operators etc. Many settings concerning the representation of characteristic curves resp. maps (colors, resolution...) are available via the main menu option OPTIONS | CUSTOMIZE... In all forms of representations the wincow content can be copied to the clipboard, printed or exported as WMF-file via the corresponding buttons.

Hint: In case that the program is *not* currently running in debug mode the characteristic map window is nevertheless available. The content, however, is only refreshed if the button of the window toolbar is clicked on.

Operation mode *characteristic curve*

In the operation mode *characteristic curve* (button) the dependency of one system output variable on one system input variable is represented in the form of a contour line. Does the system contain more than one input variable the input variables which are not selected keep their current value. In addition the current operating point is drawn in as a red bar if this option has been activated via the button.

The input variable which is to be represented is selected in the left of the two input variables comboboxes, the output variable in the right combobox. The combobox in the middle has no significance in this representation mode.



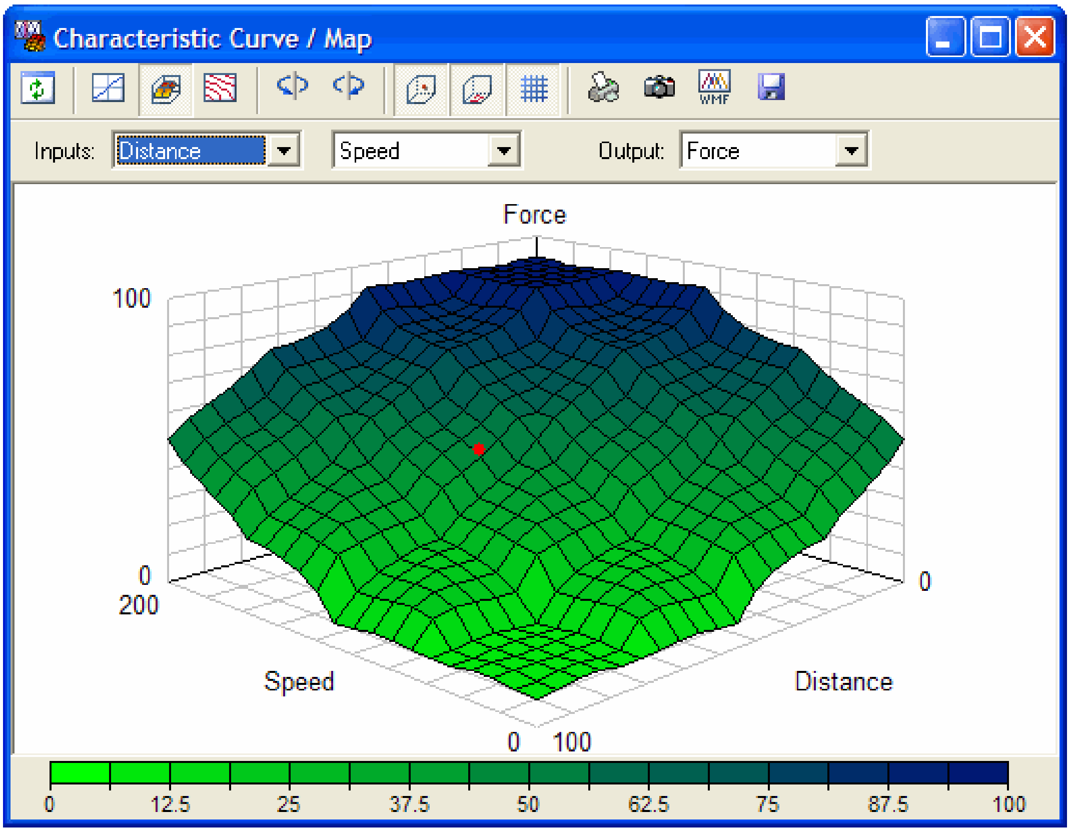
Representation of the system behaviour as characteristic curve (input variable here Distance*, output variable* Force*)*

Via the button it is possible to save the characteristic curve as a x-y-pair of values (XY file), e. g. for further use in the WinFACT module INGO.

Operation mode *characteristic map*

In the operation mode *characteristic map* (button) the dependency of one system output variable on two system input variables is represented as a 3D-characteristic map. If the system contains more than two input variables the input variables which are not selected keep their current value. In addition the current operating point is drawn in as a red point if this option has been activated via the button.

The characteristic map can be rotated by 45° steps via the resp. button. If the trace mode has been activated in the debug mode the course of the operating point is drawn in the characteristic map as a curve if this option has been activated via the button. The characteristic map can be saved as a function value matrix (FWM file), e. g. for further use in the WinFACT module INGO, via the button.

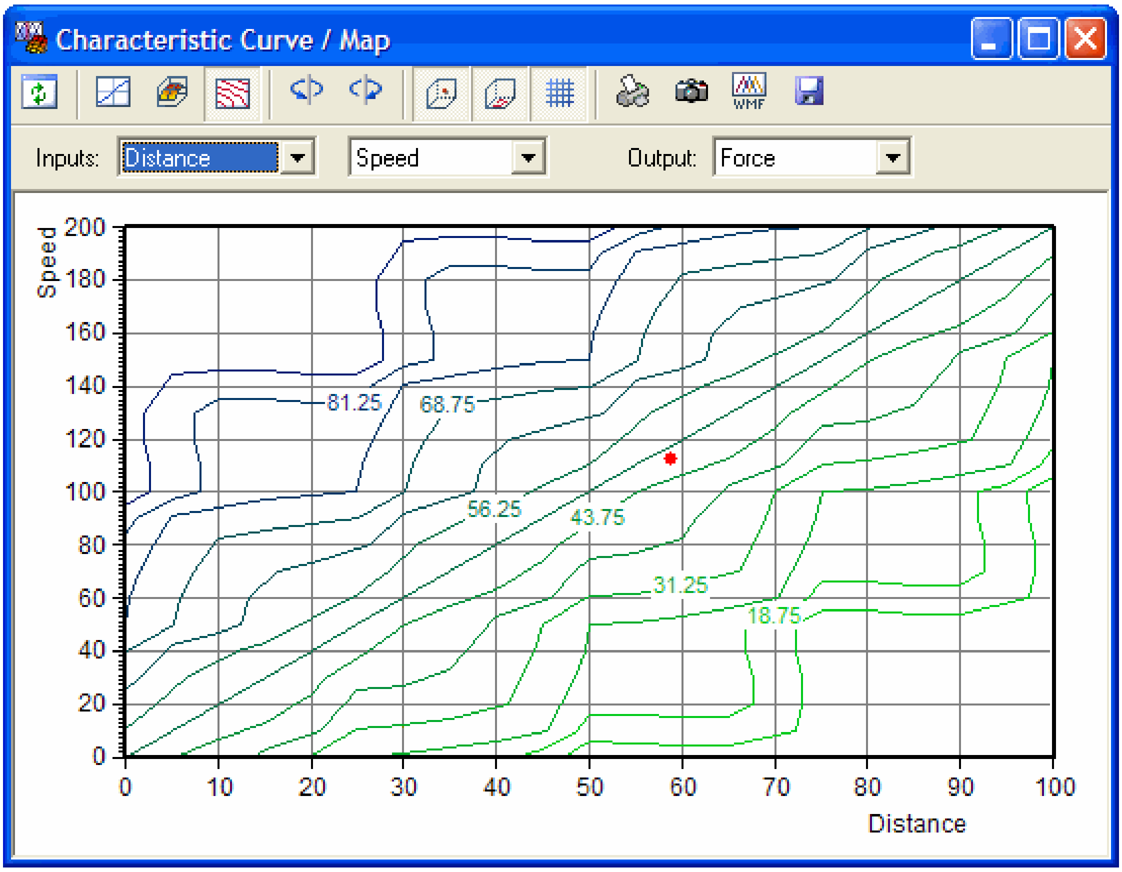


Representation of the system behaviour as 3D-characteristic map

Operation mode *contour lines*

In the operation mode *contour lines* (button) the dependency of a system output variable on two system input variables is represented in the form of contour lines, i. e. lines with a constant output variable value. If the system contains more than two input variables the input variables which are not selected keep their current value. In addition the current operating point is drawn in as a red point if this option has been activated via the button.

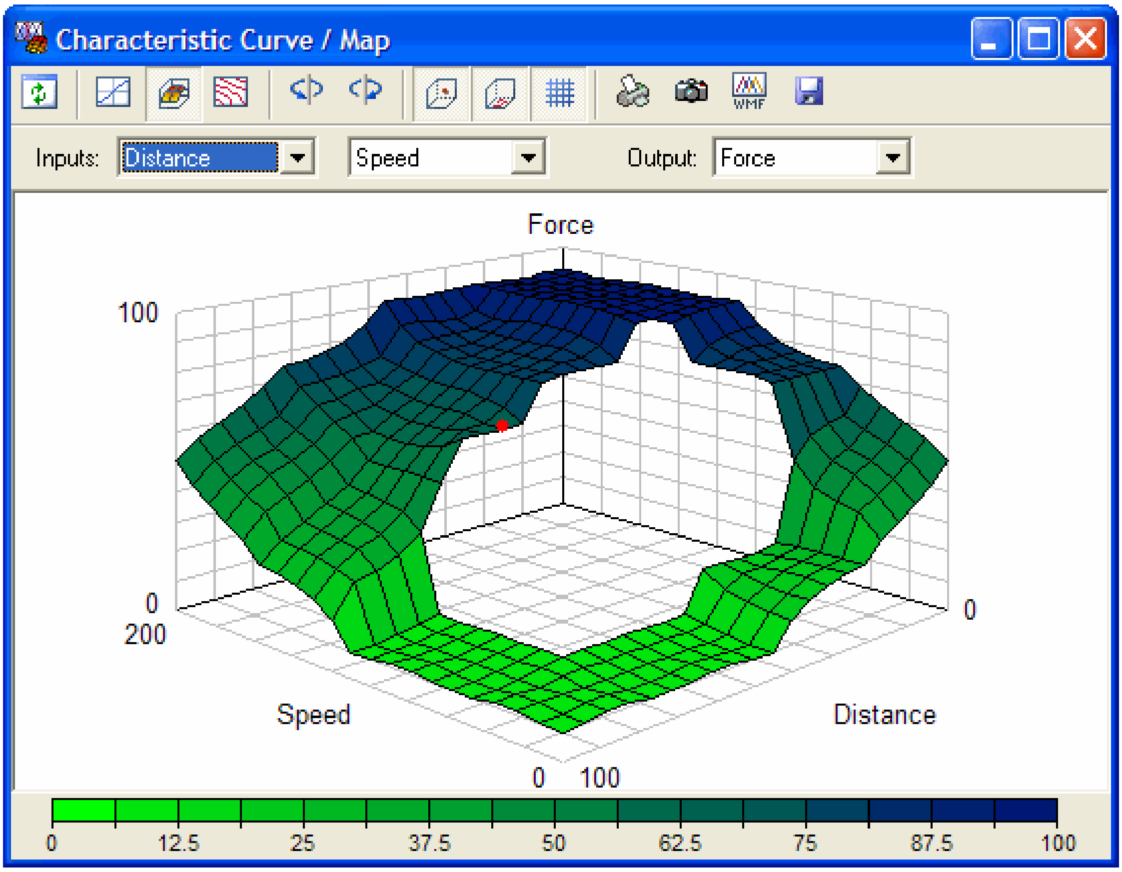
If the trace mode has been activated in the debug mode the course of the operating point is drawn in the contour lines as a curve if this option has been activated via the button. The contour lines can be saved as a function value matrix (FWM file), e. g. for further use in the WinFACT module INGO, via the button.



Representation of the system behaviour in the form of contour lines

Ambiguous states

It may occur that the calculated output value for the set input values is *ambiguous*. This ambiguousness always results if no rule is active and for this case the option *Keep old value* has been selected in the dialog for inference mechanism and defuzzification. Such areas of characteristic curves or characteristic maps are left blank. This is examplarily illustrated by the following graphic.

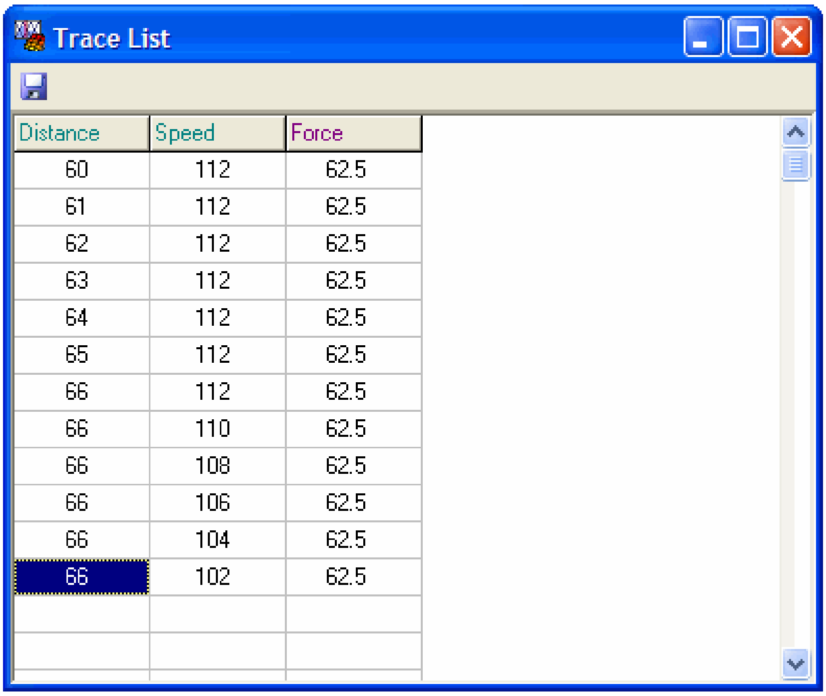


Characteristic map with ambiguous areas

The trace mode

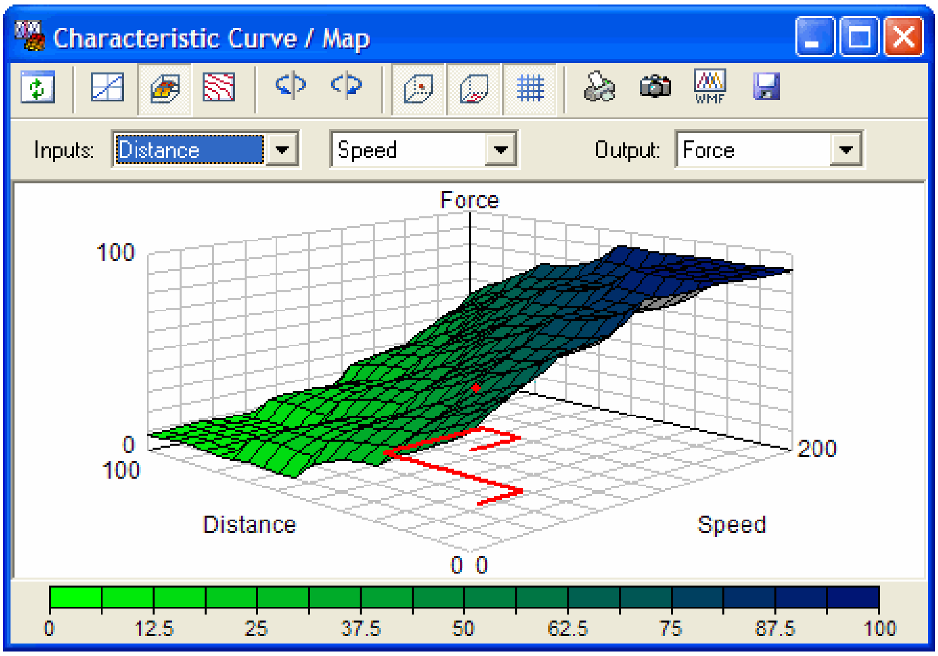
In addition to the interactive debug mode FLOP offers a trace mode which can be activated via the button. If the trace mode is activated all inference processes are recorded after having switched to the debug mode, i. e. all in- and output values of the fuzzy system are 'written down' at every change of an input value and stored to the trace buffer. This trace buffer can at any time be cleared via the button (reset trace buffer). Besides it is cleared automatically on exiting the debug mode.

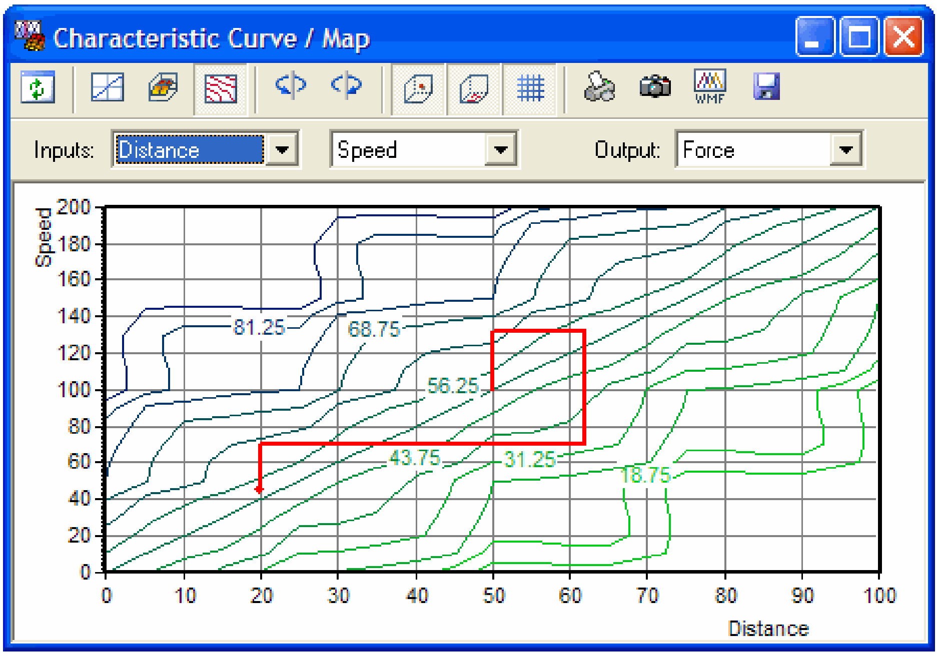
The recorded values can be displayed in tabular form in a *trace window* as it is called and can also be saved from there (e. g. for further use in other programs). The trace window can be opened via the menu option VIEW | TRACE WINDOW or the button. With closing the window the data are *not* lost but will be available after the window will be opened again as long as the debug mode was not exited. The trace mode can also be activated and deactivated several times during one debug session. In this case the recorded data are concated.



Trace window with some recorded data.
Via the button with the disk icon the data can be saved to an ASCII file.

The trace mode is of interest especially in connection with the characteristic curve/characteristic map window. If it is activated the course of the system input variables during a session or a part of it can be drawn in the characteristic map as 'trace'. By that you get important information about the operating range of the system (see the following sreenshots). This is especially interesting if the fuzzy system is running online under the blockoriented simulation system BORIS.





Characteristic map window with activated trace display in 3D mode (top) resp. contour lines (bottom)

Hint: If the debug mode has *not* been activated for the program also the trace mode has no function.

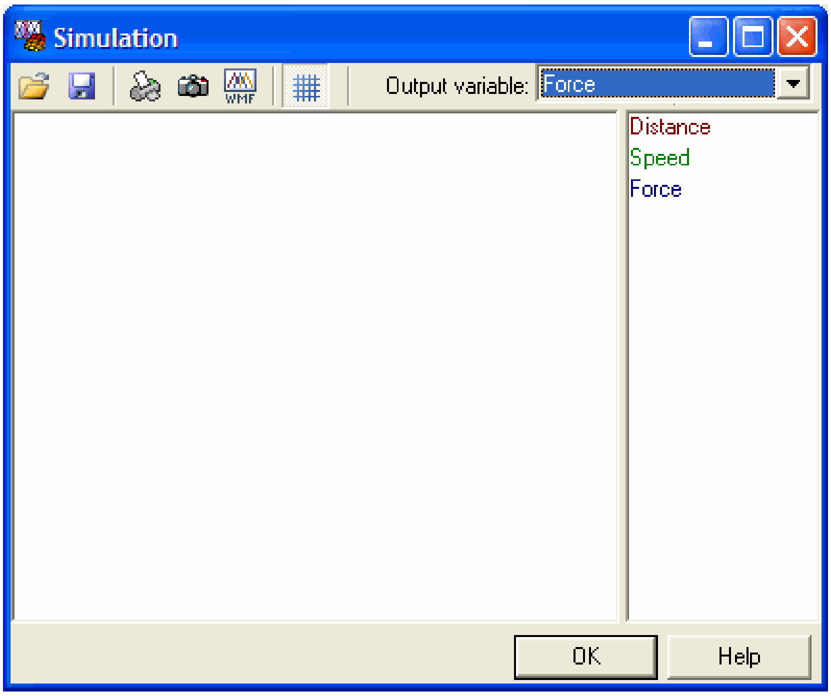
Analysis of the system by simulation

Especially if the fuzzy system is to be used as part of a control system (fuzzy controller) it is essential to analyze its behaviour by extensive simulations (if possible in combination with the controlled system). For this the fuzzy shell FLOP offers various interfaces for the WinFACT simulation module BORIS. For simple simulations there is an internal simulator which works file based.

Use of the internal simulator

Alternatively to the single step resp. characteristic map mode the transmission behaviour of a fuzzy system for any courses of the input variable(s) can be evaluated via the menu option INFERENCE | SIMULATION... or the button. In this case the input values are read from a file with the extension FSI and the resulting course of the output variable is graphically displayed. The following graphic shows the corresponding dialog before reading the simulation data. The dialog enables:

- to select the output variable which is to be represented (combobox within the window toolbar).
- to read the input values (button). The recording of the read values is shown in the display window. To represent all input courses in the same scale they are standardized to their maximum value so that all displayed values are in the range from -1 to +1.
- to print the window content (button), to copy to the clipboard (button) or to export in WMF format (button).
- to save the course of the output variable to a SIM file (button).

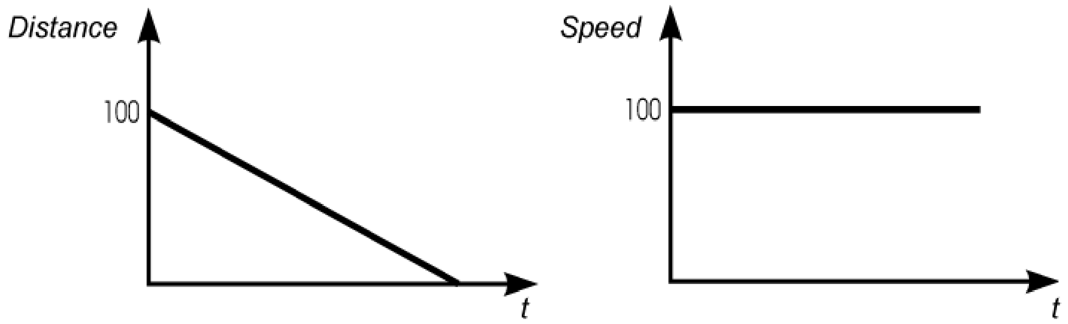


Dialog for simulation (before reading the simulation data)

The input file must contain in each row the input values for a point of time, the single values have to be separated by one or several blanks. There is no parameterization of time so that only the input values themselves have to be set. Therefore each row of a system with two input values contains two elements. The following table shows the sample data record HIGHWAY.FSI for the fuzzy system HIGHWAY.FUZ. Here the transmission behaviour of the fuzzy system is simulated for the situation that the distance decreases linearly from 100 to 0 while the speed has a constant value of 100. The following graphic shows the corresponding courses of input variables.

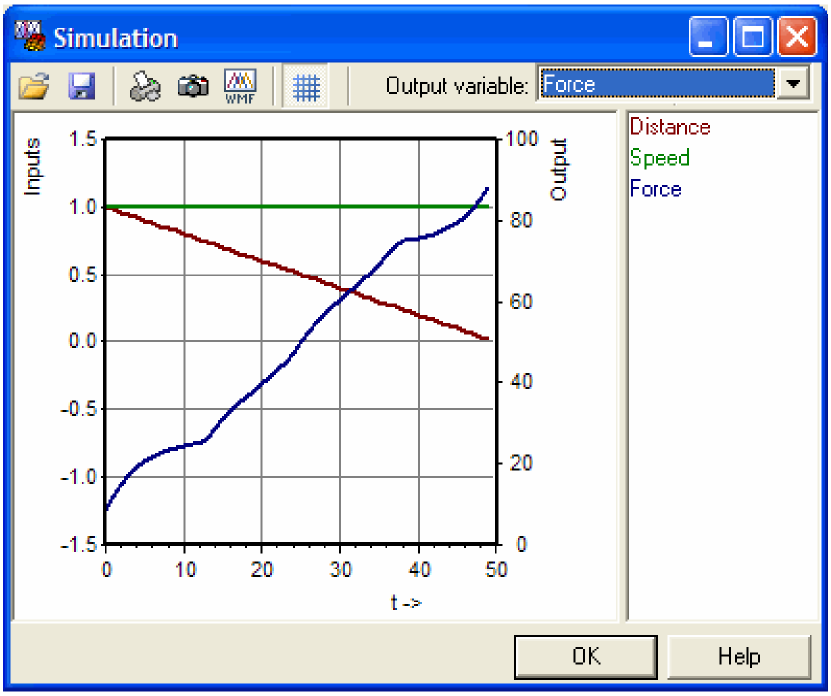
100	100
98	100
96	100
.	.
.	.
6	100
4	100
2	100

Simulation data record HIGHWAY.FSI (extract))



Courses of input variables for sample simulation

The following screenshot shows the result of this simulation.



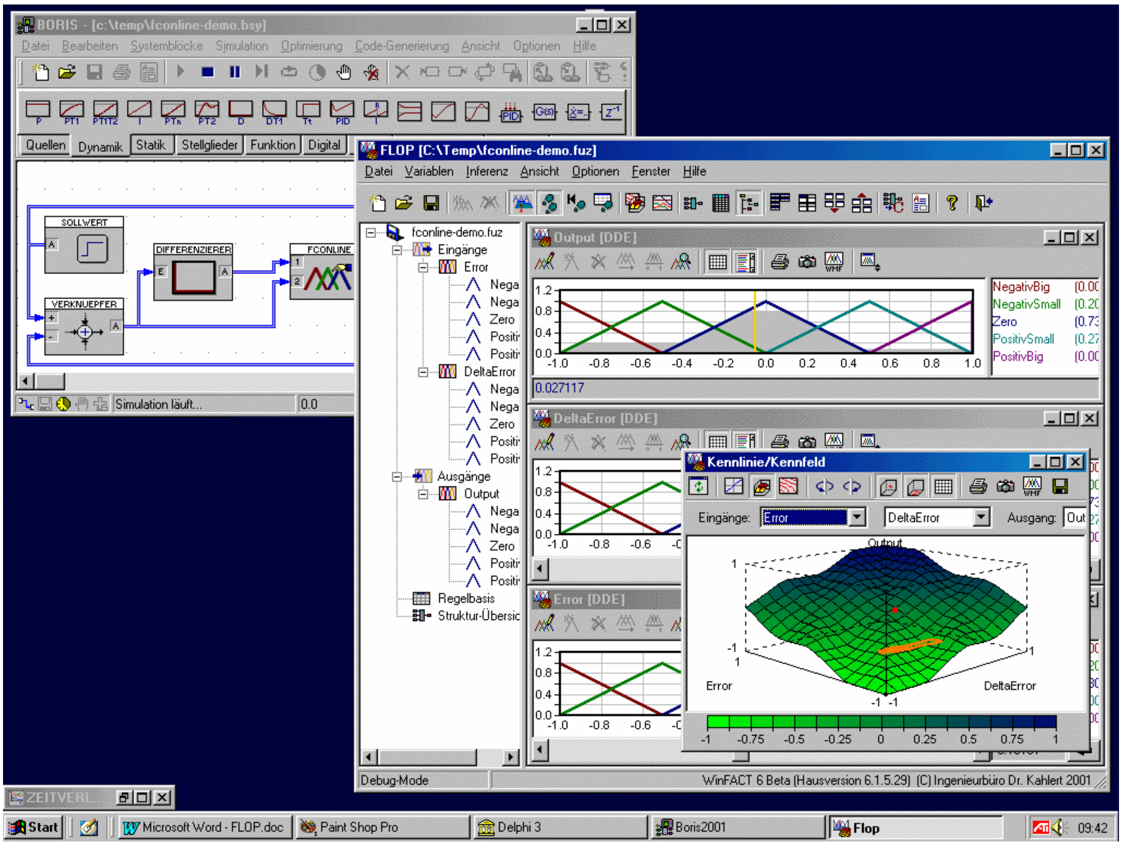
Result of the simulation for sample data record

Simulation with BORIS

Extensive simulation analyses can be carried out based on the WinFACT module BORIS. It offers two different types of blocks each of which access FUZ files generated by the fuzzy shell FLOP.

- The standard fuzzy controller (block type *FC*) which functions offline like an 'arithmetic unit'. This block type reads the selected FUZ file at the beginning of the simulation and then calculates the block output variables for each simulation step. A modification of the fuzzy controller during the simulation is *not* possible for this block type.

- The online fuzzy controller (block type *FCONLINE*). This block type is available since WinFACT 6. It starts the fuzzy shell FLOP automatically at the beginning of the simulation, loads the selected FUZ file and then communicates with FLOP during the simulation via *Dynamic Data Exchange*. By that all debug and trace functions of FLOP can be used during the simulation. Also a modification of the fuzzy controller (as far as it is allowed in the debug mode) is possible. The debug mode is automatically activated when starting the DDE connection. Further information you find in chapter 10 *Blockoriented simulation with BORIS*.



Use of the online fuzzy controller under BORIS (sample files FCONLINE-DEMO.BSY resp. FCONLINE-DEMO.FUZ): During the simulation each step of the simulation is recorded in FLOP and can be drawn e. g. in the characteristic map (here with activated trace mode)

Communication with other applications via DDE

The fuzzy shell FLOP can exchange data with other Windows applications via the standardized interface *Dynamic Data Exchange* (DDE). FLOP can work as *DDE-Server* as well as *DDE-Client*.

FLOP as DDE server

A DDE server is a program which sends data to other applications (the *DDE clients* as they are called) on their request or receives data. The DDE connection is specified by three parameters for a clear identification of the data which are to be transmitted.

Server Name of the DDE server. Normally this is the name of the executable program without the file extension EXE. If FLOP is operated as DDE server *FLOP* has to be specified as server in the DDE client program.

Topic Topic of the DDE connection. If FLOP is used as DDE server this is the name of the FUZ file without the file extension FUZ. Besides only letters, digits and underscores are allowed for *topic.* If the file name contains other characters (e. g. blanks or hyphens) they are not to be used in the specification of the topic in the DDE client.

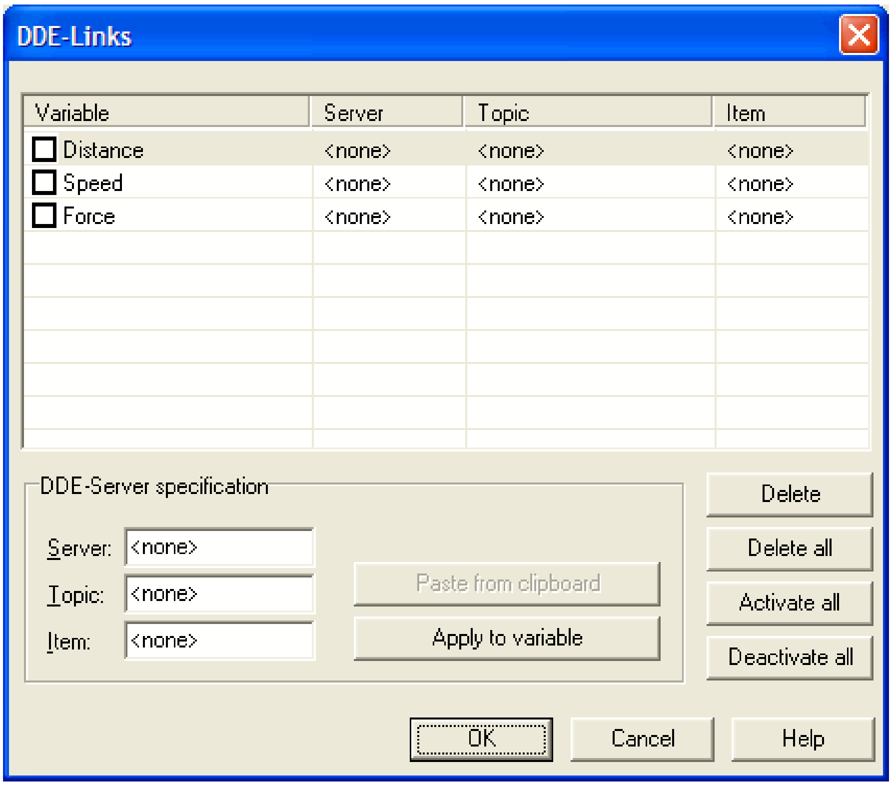
Item Data value which is to be transmitted. In FLOP this is the name of the linguistic variable which is to be addressed.

By this specification FLOP can be used as 'fuzzy artithmetic unit' which receives values for the linguistic input variables from the 'outside' via a DDE client, calculates the corresponding output values from them by the fuzzy inference and sends them back to the DDE client on request. As soon as a request is sent from a DDE client FLOP automatically switches to the debug mode. In addition the identification *[DDE]* is added in the titlebar of the variable window

The DDE server mode is used e. g. by BORIS in the FCONLINE block (see the chapter above).

FLOP as DDE client

Operating FLOP as DDE client can be used e. g. to exchange data with a spreadsheet program like EXCEL. Here EXCEL functions as DDE server. Therefore *Server*, *Topic* and *Item* have to be specified within FLOP in this operation mode. For that the menu option VARIABLES | DDE-LINKS.... can be used. In the corresponding dialog the linguistic variables and the items of the DDE server which are to exchange data can be specified.



Dialog for the specification of links for the DDE client use.

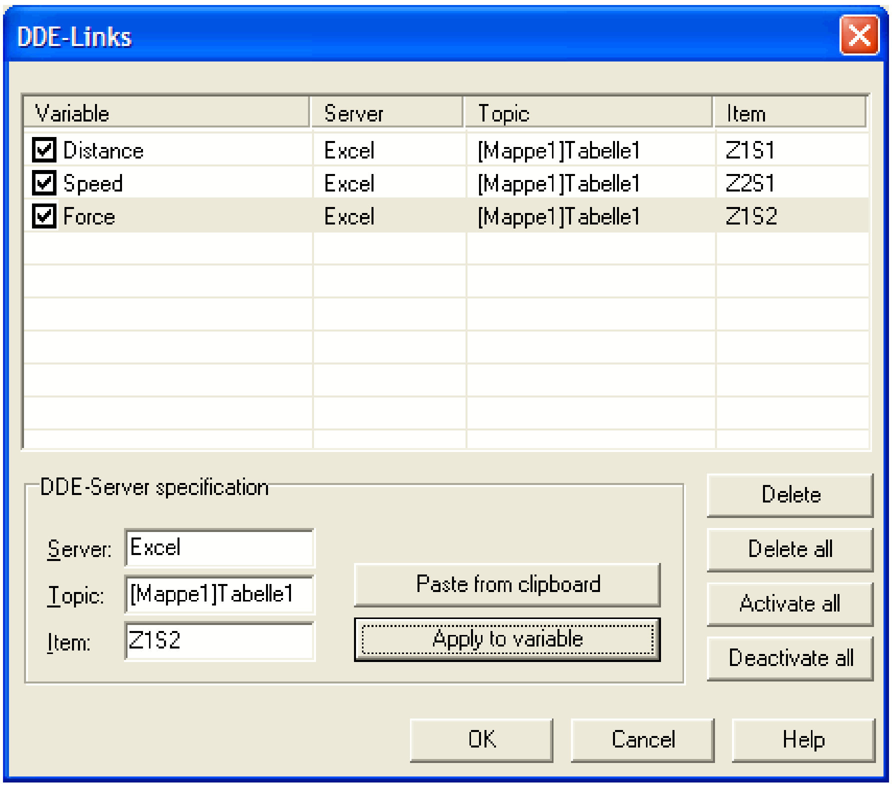
The list in the upper part of the dialog contains all linguistic variables of the fuzzy system. After clicking on a variable the corresponding link can be specified in the edit fields *Server*, *Topic* and *Item*. Via the button *Transfer to variable* the inputs are transferred and entered in the variable list. Many DDE server allow to copy a link to the Windows clipboard (see below). From there it can directly be transferred to the edit fields via the button *Paste from clipboard*.

After the new links have been defined they are automatically activated (indicated by a marked checkbox of the corresponding variable). Is the system later to be operated offline the links have only to be deactivated, they don't need to be deleted.

In the following example the sample file HIGHWAY.FUZ is used to demonstrate how a DDE connection with EXCEL is build up. For that proceed as following:

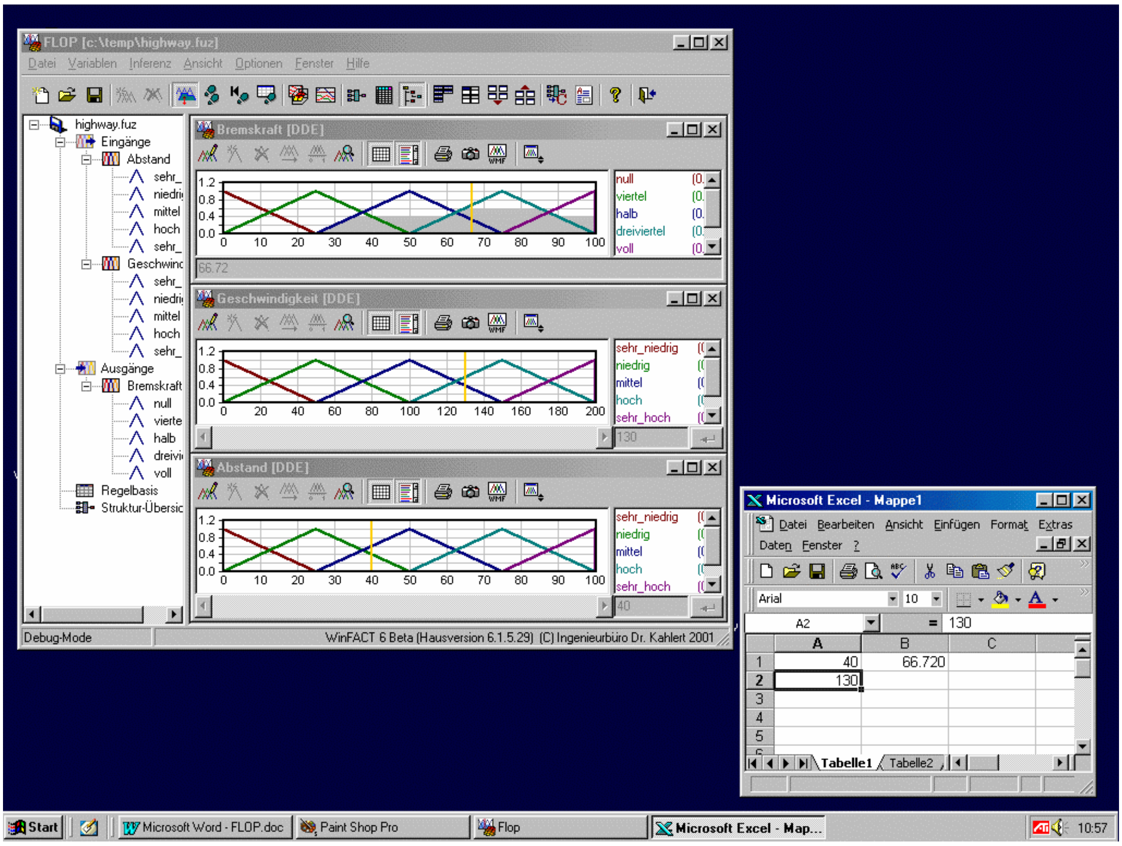
- Start FLOP and load the sample file HIGHWAY.FUZ. Switch to the dialog for the specification of the DDE links.
- Start EXCEL.
- In EXCEL click on the cell in upper left corner of the spreadsheet (A1) and select the EXCEL menu option EDIT | COPY. By that the corresponding link is copied to the Windows clipboard.
- In FLOP now click on the variable *Distance* in the dialog and in succession the buttons *Paste from clipboard* and *Transfer to variable*.
- In the same way link the variable *Speed* with the EXCEL cell A2 and the variable *Force* with the cell B1.

If everything was carried out correctly you should now see the following entries in the dialog.



Dialog after the specification of all links

After having exited the dialog the connection with EXCEL already exists and becomes activated by changing to the debug mode. In the cells A1 and A2 of the EXCEL spreadsheet the input values for the variables *Distance* and *Speed* can now be entered. They are automatically transmitted to FLOP and recorded in the corresponding variable window. For that FLOP directly calculates the corresponding output value for the *Force* and enters it in the cell B1 (see the following screenshot). On exiting the debug mode the DDE connection is automatically closed.



FLOP as DDE client for the spreadsheet program EXCEL

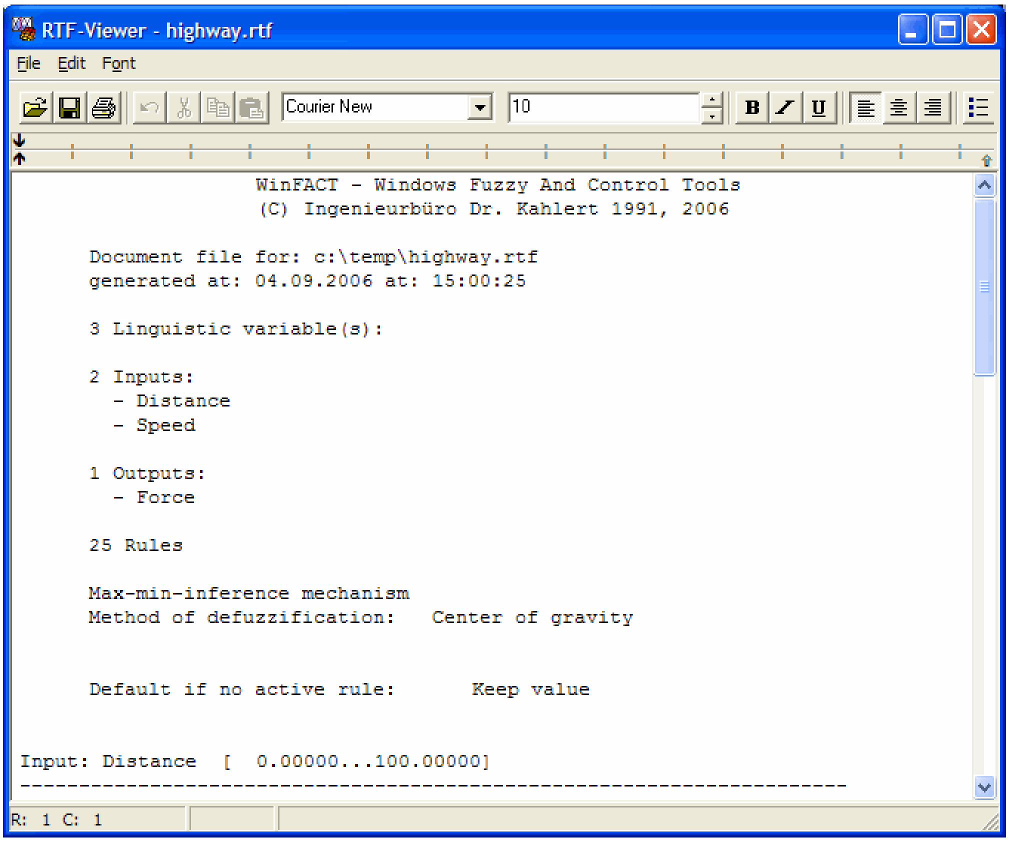
Further options

Generating C-source code

If the fuzzy C-code generator FALCO is available the actual fuzzy system can be "translated" to C-source code. For details please refer to chapter 8 *The Fuzzy-C-Code Generator FALCO*.

Generation of document file

For documentation purposes a document file in *Rich Text Format* (RTF file) can be generated via the menu option FILE | CREATE DOCUMENT FILE... or the button. This text format can be processed in all common word processing programs. FLOP already provides a simple RTF editor (*RTF viewer*) so that it is possible to print the document file directly from FLOP. After the generation of the document file the editor is automatically called and the document file already loaded. Independent of that it can at any time be displayed via the menu option VIEW | RTF VIEWER.



RTF viewer with the document file for HIGHWAY.FUZ

The following listing shows the generated document file for the file HIGHWAY.FUZ.

```
                    WinFACT - Windows Fuzzy And Control Tools
                    (C) Ingenieurbüro Dr. Kahlert 1991, 2006

     Document file for: C:\temp\highway.rtf
     generated at: 05.09.2006 at: 10:58:11

     3 Linguistic variable(s):

     2 Inputs:
       - Distance
       - Speed

     1 Outputs:
       - Force

     25 Rules

     Max-min-inference mechanism
     Method of defuzzification:   Center of gravity


     Default if no active rule:           Keep value


Input: Distance  [  0.00000...100.00000]
```

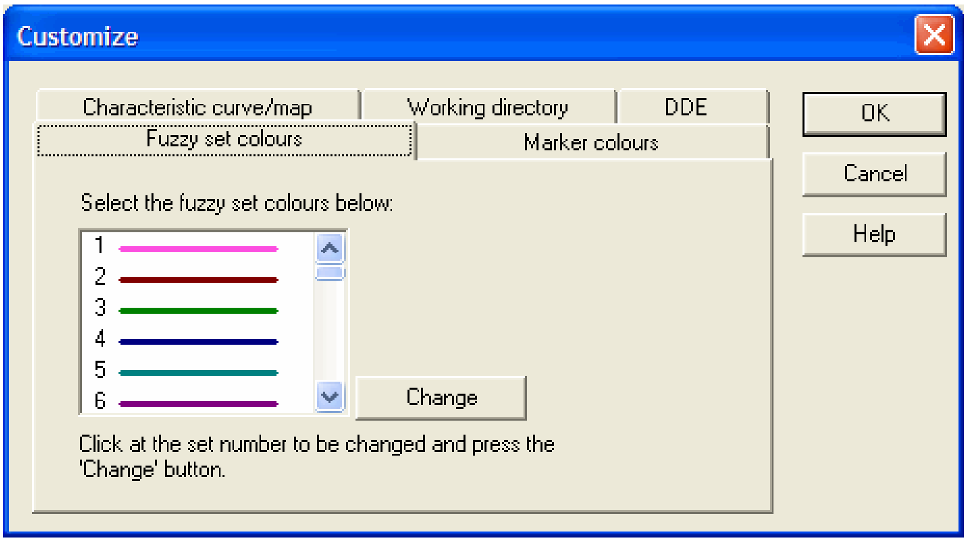
```
----------------------------------------------------------------------
Name          Form       lower limit  1. Knot    2. Knot    Upper limit
----------------------------------------------------------------------
very_low      Triangular   0.00000      0.00000              25.00000
low           Triangular   0.00000     25.00000              50.00000
medium        Triangular  25.00000     50.00000              75.00000
high          Triangular  50.00000     75.00000             100.00000
very_high     Triangular  75.00000    100.00000             100.00000

Input: Speed  [  0.00000...200.00000]
----------------------------------------------------------------------
Name          Form       lower limit  1. Knot    2. Knot    Upper limit
----------------------------------------------------------------------
very_low      Triangular   0.00000      0.00000              50.00000
low           Triangular   0.00000     50.00000             100.00000
medium        Triangular  50.00000    100.00000             150.00000
high          Triangular 100.00000    150.00000             200.00000
very_high     Triangular 150.00000    200.00000             200.00000

Output: Force  [  0.00000...100.00000; Default:   1.00000]
----------------------------------------------------------------------
Name          Form       lower limit  1. Knot    2. Knot    Upper limit
----------------------------------------------------------------------
zero          Triangular   0.00000      0.00000              25.00000
quarter       Triangular   0.00000     25.00000              50.00000
half          Triangular  25.00000     50.00000              75.00000
threequarter  Triangular  50.00000     75.00000             100.00000
full          Triangular  75.00000    100.00000             100.00000


Rule base:
---------------------------------------------------
No.  Distance  Speed      => Force        Weighting
---------------------------------------------------
1    very_low  very_low   => half           1.00000
2    very_low  low        => threequarter   1.00000
3    very_low  medium     => full           1.00000
4    very_low  high       => full           1.00000
5    very_low  very_high  => full           1.00000
6    low       very_low   => quarter        1.00000
7    low       low        => half           1.00000
8    low       medium     => threequarter   1.00000
9    low       high       => full           1.00000
10   low       very_high  => full           1.00000
11   medium    very_low   => zero           1.00000
12   medium    low        => quarter        1.00000
13   medium    medium     => half           1.00000
14   medium    high       => threequarter   1.00000
15   medium    very_high  => full           1.00000
16   high      very_low   => zero           1.00000
17   high      low        => zero           1.00000
18   high      medium     => quarter        1.00000
19   high      high       => half           1.00000
20   high      very_high  => threequarter   1.00000
21   very_high very_low   => zero           1.00000
22   very_high low        => zero           1.00000
23   very_high medium     => zero           1.00000
24   very_high high       => quarter        1.00000
25   very_high very_high  => half           1.00000
```

Document file for file HIGHWAY.FUZ

Customizing the program

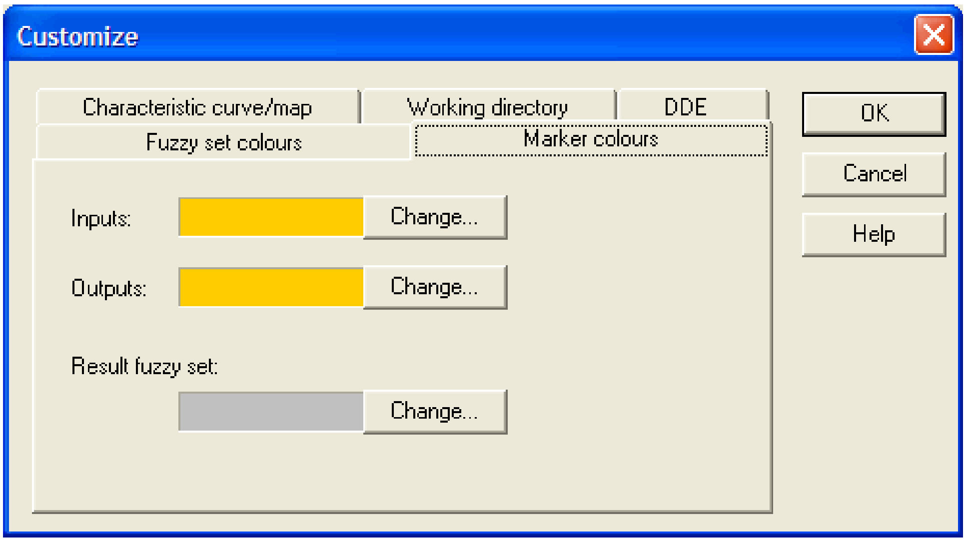
Via the menu option OPTIONS | CUSTOMIZE... the user can call a dialog which offers many options for customizing the program. All settings are saved to the Windows registry on exiting the program and are available when it is called again.

Palette *fuzzy set colors*



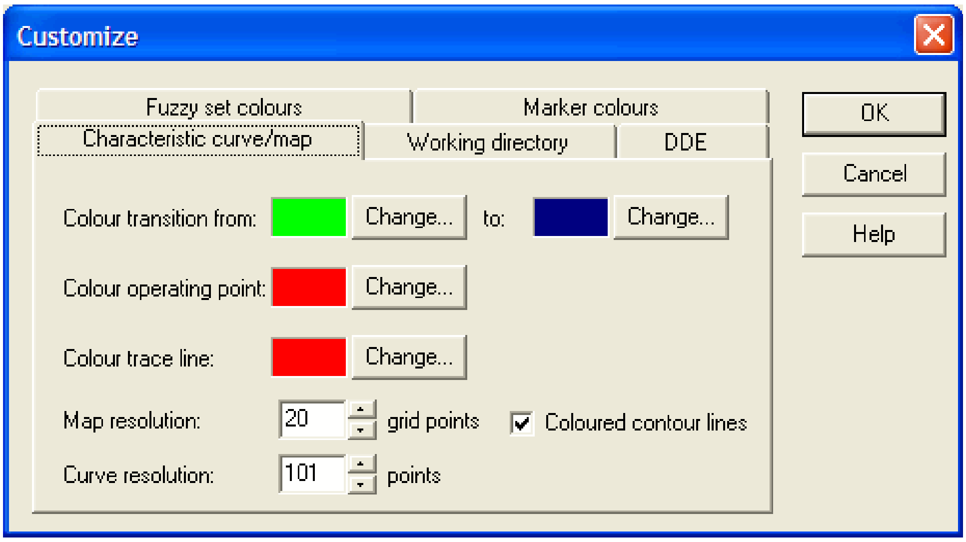
Here you specify the colors in which the fuzzy sets are drawn in their variable windows.

Palette *marker colors*



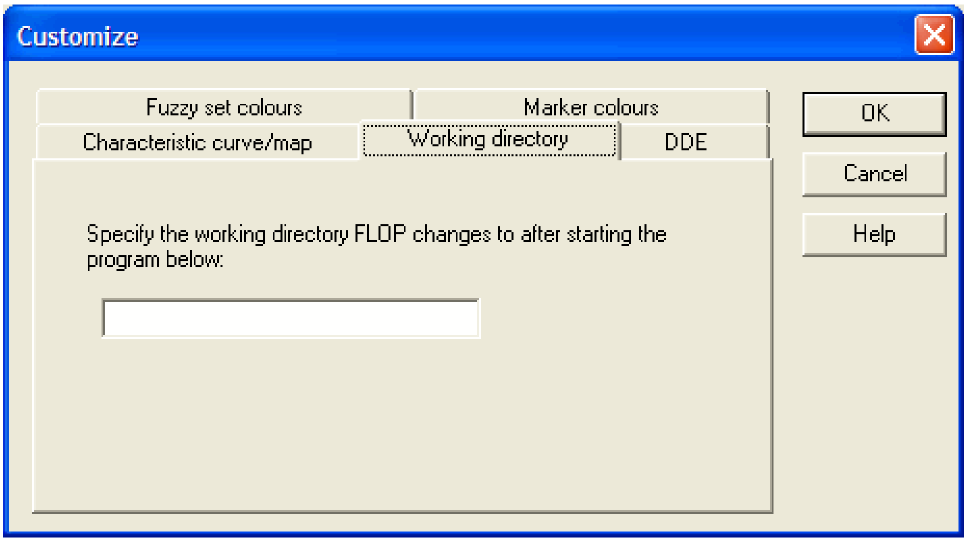
The settings on this palette specify the color of the bars (markers) by which the current values of the variables are represented in the variable window in the debug mode. The color value set via *result fuzzy set* specifies the hatching color of the result fuzzy set of the inference process for output variables.

Palette *characteristic curve/map*



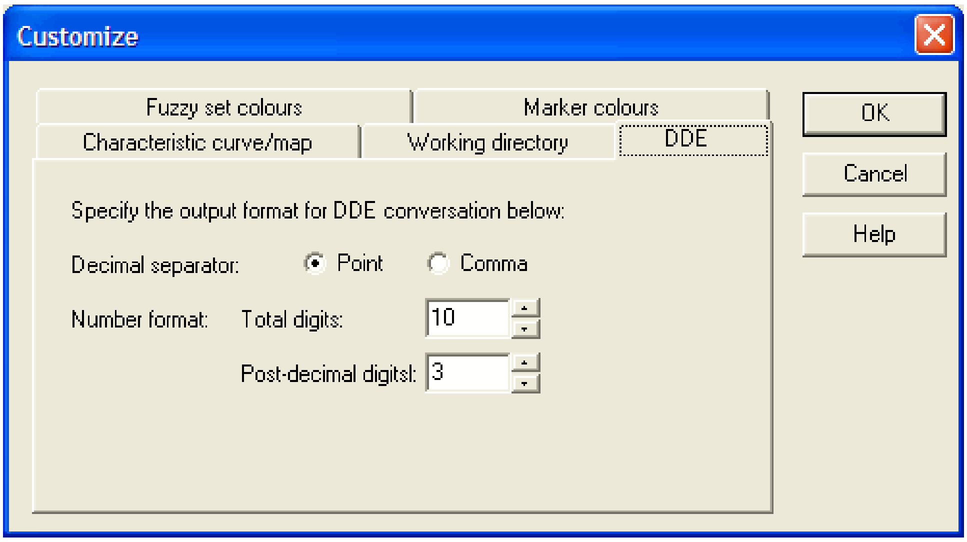
Here you can specifiy the colors for the 3D characteristic map resp. the contour lines. Besides the resolution for the characteristic map and curve can be set. Please notice that a high resolution for the characteristic map can cause very time consuming calculations.

Palette *work directory*



Here a work directory can be specified which FLOP changes to after the program start and which therefore is set as default in the *file open* dialog.

Palette *DDE*



This palette contains settings for the format of the numeric values provided by FLOP on DDE connections. By that an adjustment to various DDE client resp. server programs is possible.