

## ADAPTIVE FUZZY MODELLER

### ADVANCED DATA

- Up to 8 Input Variables and 4 Output Variables
- Up to 8 Fuzzy Sets for each Input Variables
- Up to  $2^{14}$  Fuzzy Rules
- Rules Learning Phase using an unsupervised WTA-FAM
- Membership Functions Learning Phase using a supervised BACK-FAM
- Automatic and Manual Learning Rate
- Rules Minimizer
- Gaussian and Triangular Membership Functions Shape
- Inference method based on Product or Minimum
- Step-by-Step and from File Simulation available
- Supported Target: W.A.R.P. 1.1, W.A.R.P. 2.0, MATLAB and ANSI C

### DESCRIPTION

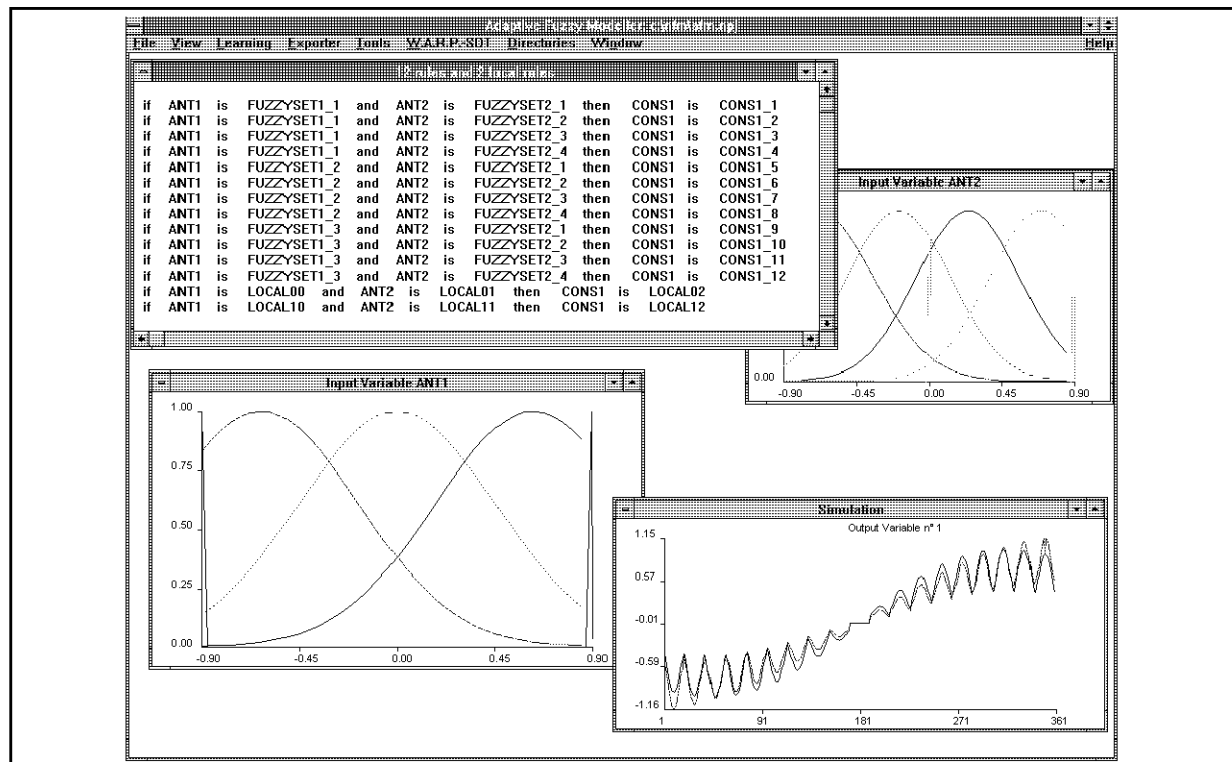
Adaptive Fuzzy Modeller (AFM) is a tool that easily allows to obtain a model of a system based on Fuzzy Logic data structure, starting from the sampling of a process/function expressed in terms of Input/Output values pairs (patterns).

Its primary capability is the automatic generation of a database containing the inference rules and the parameters describing the membership functions. The generated Fuzzy Logic knowledge base represents an optimized approximation of the process/function provided as input.

The AFM has the capability to translate its project files to FUZZYSTUDIO™ project files, MATLAB and C code, in order to use this environment as a support for simulation and control.

The block diagram in fig.2 illustrates the AFM logic flow.

**Figure 1. Block Diagram**



### LEARNING

it is composed by two phases:

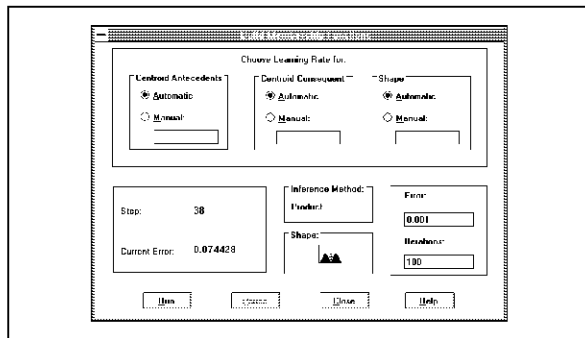
**BUILDING RULES** It allows to perform the automatic selection of inference rules or their manual definition, taking in to account the project constraints read from the previously opened pattern file.

As a result the user will be supplied with a rule file containing the linguistic expression of the rules. An unsupervised clustering algorithm is used to perform this task.

**BUILDING MEMBERSHIP FUNCTIONS** It allows the user to select the membership function shape and the fuzzy inference method for the project elaboration.

Starting from the rule file supplied by the previous phase, it initially associates to each fuzzy set a standard membership function shape. These shapes can be gradually tuned in order to let the fuzzy system to better approximate the process/function sampling by means of subsequently run sessions. Back-propagation algorithm with automatic learning rate control is used to this aim.

**Figure 3. BUILD MEMBERSHIP FUNCTION window**



### TOOLS

It is composed of different sub-menus:

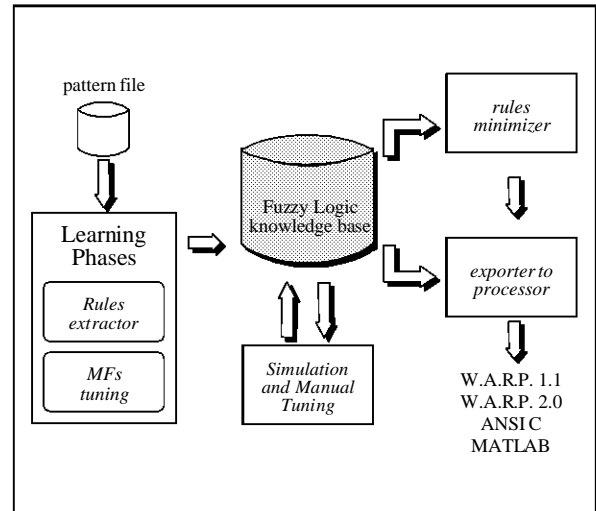
**LOCAL RULES** it allows to add new rules to the fuzzy logic knowledge base determined by an Adaptive Fuzzy Modeller run session. Aim of this functionality is the local approximation level improvement.

**SIMULATION** it allows to simulate the fuzzy system behaviour in order to verify the approximation level obtained during the learning phase. The simulation can be carried out in two different ways.

**Simulation Step-by-Step:** the user must supply the simulator with the values variables corresponding to the point to verify.

**Simulation from File:** the user must supply the simulator with the name of a process/function stream file that will be used to perform a complete process inference.

**Figure 2. AFM Logic Flow.**



### VIEW FEATURES

View Features of the AFM gives with the capability to visualize the fuzzy model extracted for a particular project. It allows a separate visualization of the rules of inference and membership functions. The rules can be visualized in a linguistic format. For the membership functions you can choose between a linguistic and a graphical format visualization.

### EXPORTERS

The Exporter provides library functions working on the databases automatically generated, which appropriately describe the data structures of the selected project in terms of a different programming environment.

These functions can be exploited inside the user's programs in order to verify the model extracted and to use it in real application.

### SUPPORTED TARGETS

The supported environment are:

- W.A.R.P.1.1 using FUZZYSTUDIO™1.0
- W.A.R.P.2.0 using FUZZYSTUDIO™2.0
- MATLAB
- C Language
- Fu.L.L. (Fuzzy Logic Language).

# SYSTEM REQUIREMENTS

MS-DOS version 3.1 or higher  
 Microsoft Windows 3.0 or compatible later version  
 486, PENTIUM compatible processor chip  
 8 MBytes RAM (16 Mbytes recommended)  
 Hard Disk with at least 1MBytes free space

Order Code	Description	Supported Target	Functionalities	System Requirement
STFLAFM10/SW	WTA-FAM for Building Rules BACK-FAM for Building MFs	STFLWARP11/PG STFLWARP11/PL STFLWARP20/PL ANSI C MATLAB®	Rules Minimizer Step-by-Step Simulation Simulation from File Local Tuning	MS-DOS 3.1 or higher Windows 3.0 or later 486, PENTIUM compatible 8 MB RAM

Type	Description	Operating Temperature	Package
STFLWARP11/PG-PL	HCMOS, 6KBytes RAM, 40MHz, 16 Inputs, 16 Outputs, 256 Rules	0 - 70°C	CPGA100-PLCC84
STFLWARP20/PL	HCMOS, 1.4KBytes RAM, 40MHz, 8 Inputs, 4 Outputs, 256 Rules	0 - 70°C	PLCC68

Type	Device	Development Tools	
		FUZZYSTUDIO™ ADB	FUZZYSTUDIO™ SDT
STFLSTUDIO10/KIT	STFLWARP11/PG	W.A.R.P. 1.X W.A.R.P. 1.X programmer EPROM programmer RS-232 communication handler Internal Clock	Variables and Rules Editor W.A.R.P. Compiler/Debugger Exporter for ANSI C and MATLAB®
STFLSTUDIO20/KIT	STFLWARP20/PL	W.A.R.P. 2.0 W.A.R.P. 2.0 programmer ZEROPOWER programmer RS-232 communication handler Internal Clock	Variables and Rules Editor W.A.R.P. Compiler/Debugger Exporter for ANSI C and MATLAB®

Information furnished is believed to be accurate and reliable. However, SGS-THOMSON Microelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of SGS-THOMSON Microelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. SGS-THOMSON Microelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of SGS-THOMSON Microelectronics.

© 1996 SGS-THOMSON Microelectronics – Printed in Italy – All Rights Reserved

FUZZYSTUDIO™ is a trademark of SGS-THOMSON Microelectronics  
MS-DOS®, Microsoft® and Microsoft Windows® are registered trademarks of Microsoft Corporation.  
MATLAB® is a registered trademark of Mathworks Inc.

SGS-THOMSON Microelectronics GROUP OF COMPANIES  
Australia - Brazil - Canada - China - France - Germany - Hong Kong - Italy - Japan - Korea - Malaysia - Malta - Morocco - The Netherlands -  
Singapore - Spain - Sweden - Switzerland - Taiwan - Thailand - United Kingdom - U.S.A.