UNIVERSITY of OULU

CONTROL ENGINEERING LABORATORY
Department of Process Engineering

Fuzzy Modelling with Linguistic Equations

Ari Isokangas
Esko Juuso

Report A No 11, February 2000

University of Oulu
Department of Process Engineering
Control Engineering Laboratory
Report A No 11, February 2000

Fuzzy Modelling with Linguistic Equations

Ari Isokangas and Esko Juuso

Control Engineering Laboratory,
Department of Process Engineering,
University of Oulu
P. O. Box 4300, FIN-90014 University of Oulu, Finland
Email: {ari.isokangas | esko.juuso}@oulu.fi

Abstract
In this report, different types of fuzzy models have been developed from linguistic equations models. Different shapes of membership functions were compared: triangular and trapezoidal membership functions as well as their non-linear modifications were used. ANFIS (adaptive neuro-fuzzy inference system) method for Takagi-Sugeno type models was used and clustering for Singleton fuzzy models. Also the different number of singleton values in singleton fuzzy models and by using fuzzy relations different amount of rules was compared. The data-based approaches are based on data from the Cooking Liquor Analyser CLA 2000. Linguistic equations (LE) work well for this data. For the test data, the performance of the real-valued LE model was the best although a better fitting accuracy with training data was obtained by constructing Takagi-Sugeno (TS) fuzzy models with the ANFIS method. There are also overfitting problems with the TS models. Easy configuration and robustness are the main benefits of the LE models. The fitting performance must be compared to the number of modelling parameters.

Keywords: fuzzy models, fuzzy logic, linguistic equations, linguistic relations, ANFIS.

ISBN 951-42-5546-1
ISSN 1238-9390
ISBN 951-42-7506-3 (PDF)

University of Oulu
Department of Process Engineering
Control Engineering Laboratory
PL 4300
FIN-90014 University of Oulu

TABLE OF CONTENTS

1 INTRODUCTION

Proper control of the pulp digester is very important to the pulp production. In the continuous digester, raw materials containing cellulose fibres are cooked with chemicals (cooking liquor) in high temperature (about 150 – 170 ^{o}C, depending on the wood species and grade requirements). While the cooking progresses, some wood species start to dissolve into the cooking liquor. The aim is to remove enough lignin from wood to get the fibres free. In sulphate pulping the cooking liquor is alkaline. Usually alkali components of the liquor are analysed by measuring the conductivity of the liquor sample, solid contents by measuring the refractive index, and dissolved lignin by measuring the UV-absorbency of the sample (Haataja et al., 1997). ABB's Cooking Liquor Analyser (CLA 2000) is an advanced measurement device developed for analysing the chemical pulping process.

With these measurements it is possible to estimate the development of the Kappa number long before the end of the cook. Different approaches have been used for mathematical modelling of the cooking result (Murtovaara et al., 1999): fuzzy logic (FL), partial least squares method (MLR), artificial neural networks (ANN) and linguistic equations (LE). Measurements of the CLA 2000 system and on-line Kappa-number measurements were used in the development of these models. The models have three input variables: refractive index, conductivity and temperature. Temperature has the biggest effect on output. These models can be used in cooking control to reduce the Kappa-number variation.

Neural network models and linguistic equation models seem to learn the process behaviour in a similar manner. Differences come out in using the models in process environment (Murtovaara et al., 1999). Neural network models are suitable for processes where process conditions are very stable and there is a lot of data available. As the linguistic equation model is not too sensitive for changes in process conditions, it is better and more suitable for the Kappa-number prediction.

Different approaches and development methods can be combined through the Linguistic Equation Framework, which provides a unified method for developing and tuning adaptive fuzzy expert systems (Juuso, 1999a). The modelling can be done with *FuzzEqu Toolbox* created in *Matlab*® environment. Automatic data analysis is started with generating the membership definitions from the data (Juuso, 1999b). Linguistic equations are developed from real valued relations obtained from the data by non-linear scaling based on the membership definitions. In small systems, the directions are usually quite clear: only the absolute values of the coefficients need to be defined. For more complex systems, a set of alternative equations is developed first, and the final set of equations is selected on the basis of error measures and process knowledge. Membership definitions can be tuned for selected variables: the variable alternatives are restricted by the equation set.

The *FuzzEqu Toolbox* contains routines for modifying membership definitions interactively: definitions can be extended or contracted independently on positive or negative or on both sides; definitions can be moved as well. All proposed changes are checked in the toolbox. With these tools the system can be adapted to changing operating conditions. Fuzzy models can be changed into linguistic equation models by replacing linguistic labels with real numbers (Juuso, 1999a). The *FuzzEqu Toolbox* includes routines for building a single LE system from large fuzzy systems including various ruleblocks implemented in *FuzzyCon* or *Matlab* *Fuzzy Logic Toolbox*. Other fuzzy modelling approaches can be used as channels for combining different sources of information.

Fuzzy modelling suits very well to multivariable non-linear modelling. Fuzzy models can be constructed from expertise and data. Fuzzy models on any fuzzy partition can be generated from LE models: rules or relations are developed either sequentially or simultaneously (Juuso, 1999a), and membership functions are generated from the membership definitions on any location. Every equation can define the locations of membership functions for one selected variable. Triangular membership functions suit very well for linguistic fuzzy models and singleton models. For Takagi-Sugeno (TS) fuzzy models, trapezoidal membership functions or smoother non-linear versions are better (Juuso, 1999b).

Singleton models represent the LE model quite accurately if the locations of the membership functions are based on the shapes of the membership definitions in such a way that the linear surfaces are on appropriate areas. Takagi-Sugeno (TS) fuzzy models have the same requirement, but the locations of the membership functions are different. Linguistic fuzzy models are developed from singleton models. Fuzzy relational models are useful for development of fuzzy LE models. There will be fairly few nonzero elements since non-linearities are included to the membership functions.

This report discusses a Kappa-number estimator using different types of fuzzy models and linguistic equations. Fuzzy models were developed from linguistic equations using different shape and different number of membership functions. These models are compared to fuzzy models developed by neuro-fuzzy ANFIS method.

2 INTELLIGENT MODELLING

Fuzzy modelling is an extension of the expert system techniques to uncertain and vague systems. Fuzzy set systems continue the traditions of physical modelling on the basis of understanding the system behaviour. Fuzzy rules and membership functions can represent gradually changing non-linear mappings together with abrupt changes. Fuzzy models can help in extracting expert knowledge on an appropriate level (Juuso 1996). Fuzzy models can also be constructed from data, which alleviates the knowledge acquisition problem. Various techniques have been used to fit the data with the best possible accuracy, but in most cases the interpretation of results is not addressed sufficiently. Fuzzy models can also be considered as a class of local modelling approaches, which attemps to solve a complex modelling problem by decomposing into number of simpler subproblems (Babuska 1997).

Fuzzy modelling is usually based on rule-based models. The most common alternatives are linguistic fuzzy models (Driankov et al., 1993), which suit well extracting expert knowledge. Takagi-Sugeno fuzzy models are suitable for constructing systems from data (Takagi and Sugeno, 1985). Singleton fuzzy models can be considered as an intermediate approach between these two. Fuzzy relational models also provide a technique for extracting models from data. Linguistic equations (LE) can be used as an unified framework for combining these modelling techniques (Juuso, 1999a). The LE approach facilitates also adaptation of fuzzy set systems to changing operating conditions.

2.1 Linguistic fuzzy models

Linguistic fuzzy models, where both the antecedent and the consequent are fuzzy propositions, can be interpreted by using natural language, heuristics and common sense knowledge. Linguistic fuzzy if-then rule can be represented in a general form:

$$R_i: \text{ If } x \text{ is } A_i \text{ then } y \text{ is } B_i,$$

where x is the antecedent variable (input). Similarly, y is the consequent variable (output). A_i and B_i are linguistic terms (labels), which are defined by fuzzy sets

$$\mu_{Ai}(x): X \rightarrow [0,1] \text{ and } \mu_{Bi}(y): Y \rightarrow [0,1].$$

The antecedent propositions contain fuzzy sets directly defined in the vector domain X. More often, the rules are in a decomposed form where the antecedent is defined as a combination of simple fuzzy propositions on the individual components x_i, i=1, 2,…, p of the vector x. By using the logical operators (conjunction, disjunction and negation) a desired compound proposition can be constructed. For example

$$R_I: \text{ If } x_i \text{ is } A_{i1} \text{ or } x_2 \text{ is } A_{i2} \text{ and } x_3 \text{ is not } A_{i3} \text{ then } y \text{ is } B_i,$$

where the degree of fulfilment of the rule is calculated using the appropriate t-norm, t-conorm and the complement operator.

The inference mechanism is based on the generalized *modus ponens* rule:

If *x* is A then *y* is B
x is A'
----
y is B'

"If *x* is A then *y* is B" and "x is A' ", an output fuzzy set B' is derived by:

$$B' = A' \circ R.$$

Here ° denotes the sup-t composition. For the minimum t-norm and the maximum t-conorm, the max-min composition is defined by

$$\mu_{B'}(y) = \max_{x} \min_{x,y}(\mu_{A'}(x), \mu_R(x,y)).$$

The result of linguistic fuzzy models is the fuzzy set B'. If a numerical output value is required, the output fuzzy set must be defuzzified.

Degrees of freedom in linguistic fuzzy models are fairly high since all membership functions and rules can be defined independently. However, some constraints are preferable for practical tuning of the systems, e.g. in fuzzy control the membership functions are defined in such a way that no more than two functions are active at a same time for any variable.

Rule generation is usually based on membership functions defined in the procedure (Juuso, 1999a). Table-lookup scheme (Wang and Mendel, 1992; Wang, 1994) is a one-pass procedure for generating fuzzy rules from numerical I/O-data with capability to combine linguistic information into a common rule base. This methodology does not offer any means to identify the structure of the system. Fuzzy rule generation can also take into account contradictory data (Krone and Kiendl, 1994; Krone and Schwane, 1996).

2.2 Singleton fuzzy models

In singleton fuzzy models, the consequent fuzzy sets B_i of a linguistic model can be reduced to fuzzy singletons and represented as real numbers b_i:

Ri: If x is A_i then $y = b_i$

When singleton fuzzy model is compared with linguistic fuzzy model, the number of distinct singletons in the rule base is usually not limited. It means that each rule may have its own singleton consequent.

2.3 Takagi-Sugeno models

Takagi-Sugeno models are suitable for approximation of a large class of non-linear systems. In Takagi-Sugeno (TS) fuzzy model, the rule consequents are crisp functions of the model inputs:

$$R_i:\ \text{If } x \text{ is } A_i \text{ then } y_i = f_i(x), \quad i=1,2,\dots,K,$$

where X is input and y is output variable. R_i denotes the *i*th rule, and K is the amount of rules in the rule base.

The consequent functions f_i are usually chosen as instances of a suitable parameterised function, whose structure remains equal in all the rules and only the parameters change. An affine linear form is simple and practically useful in parameterisation:

$$y_i=a^T x + b_{i,}$$

where a_i is a parameter vector and b_i is a scalar offset. The previous model can be called as an affine TS model. A special case occurs when offset $b_i=0$. In such a case the model can be called a homogenous TS model. This kind of model has not so good approximation capabilities than the affine TS model. If the parameter vectors $a_i=0$, i=1,….,K, the consequents of the model are constant functions and the model is called singleton model.

Takagi-Sugeno fuzzy models can generated from data by hybrid neural nets. The neural net defines the shape of the membership functions of the premises. This architecture and learning procedure is called ANFIS (adaptive neuro-fuzzy inference system) (Jang, 1997). The approach uses input/output data set to construct a fuzzy inference system (FIS), whose membership function parameters are adjusted using either a backpropagation algorithm alone or in combination with least square type of method. This allows fuzzy systems to learn from the data they are modelling. ANFIS method is used for deloping Takagi-Sugeno type fuzzy models.

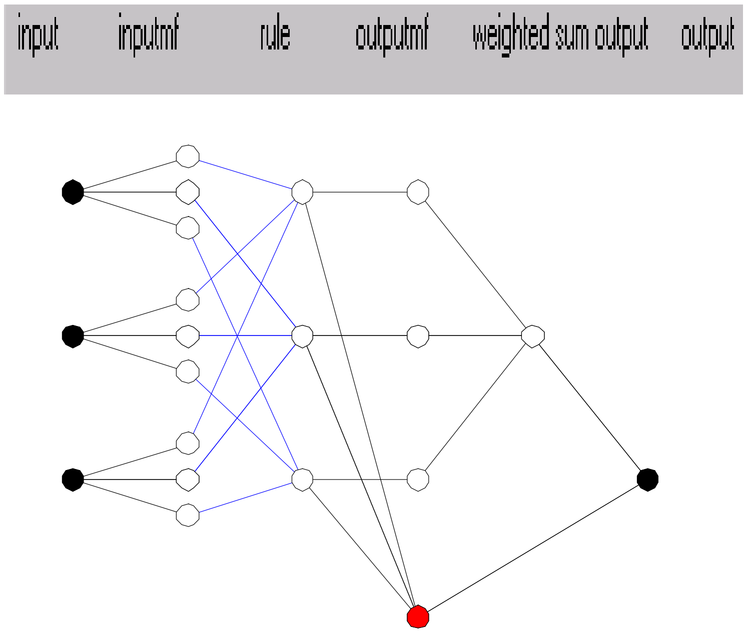

Figure 1. ANFIS architecture for Takagi-Sugeno type system.

2.4 Fuzzy relational models

Fuzzy relational models (Pedrycz, 1984) describe associations between linguistic terms defined in the input and output domains of the system. The individual elements of the relation represent the strength of association between the fuzzy sets. Denote A a collection of M linguistic terms (fuzzy sets) defined on domain X, and B a collection of N fuzzy sets defined on Y:

$$A = \{A_1, A_2, ..., A_M\},$$
$$B = \{B_1, B_2, ..., B_N\}.$$

The model is a table storing the rule base in which all the antecedent combinations are tied to all the consequents with different weights.

To obtain a crisp output, the resulting fuzzy set is defuzzified by the fuzzy-mean method applied to the centroids b_j of the fuzzy sets B_j. The centroids b_j can be calculated using different defuzzification methods, e.g. center of gravity. The crisp output of the fuzzy relational model is calculated by a weighted mean of the centroids.

A linguistic fuzzy model can be regarded as a special case of the fuzzy relational model. In this case, relation R is a crisp relation constrained such that only on nonzero element is allowed in each row of R, i.e. each has only one consequent. Fuzzy relational models can also be represented as singleton fuzzy models if centroids of the output membership functions are defined in advance.

2.5 Linguistic equations

Linguistic equations can be shown in the form

$$\sum_{j=1}^{m} A_{i,j} X_j + B = 0,$$

where X_j is a linguistic level for the variable j. Levels X_j are calculated by membership definitions from measurements of the corresponding variable j. B is a bias term. Interaction matrix A describes interactions between variables. The interaction matrix consists of real valued coefficients which can be defined from the training data.

Each equation represents a multivariable interaction: directions and strengths of interactions are defined by the coefficients of the interaction matrix. Only the variables with a nonzero coefficient belong to the interaction. As the equations are linear, very large systems can be packed efficiently. Non-linearities are handled with membership definitions which transform the measurements into the range [-2, 2]. This approach can be considered as a non-linear scaling technique which linearises the system. Since only five parameters are needed for each variable, the LE system can be adapted to various operating conditions.

The membership definitions consist of two polynomial functions, one for the linguistic values [-2, 0] and one for the linguistic values [0, 2]. The functions are connected at the linguistic value zero. The polynomial functions are developed from the training data by fitting a second order polynomial against the data points (Juuso, 1999b). The polynomials have to be selected so that the membership definition is always monotonously increasing. By membership definitions we convert measured signals into the range corresponding linguistic values from very low to very high. This action combines scaling and fuzzification.

Usually, fuzziness is taken into account by membership definitions – linguistic equation approach does not necessarily require any uncertainty or fuzziness. However, linguistic equations can be used in fuzzy form as well.

Fuzzy models on any partition can be generated from linguistic equation models. For linguistic and singleton fuzzy models, this is done by solving the set of equations after selecting the locations of membership functions for n-1 variables if the equation or the subset of equations includes n variables. The locations of membership functions for one variable in each subset are obtained from the equation model. Number of output membership functions is restricted by selecting a suitable threshold for differences between locations of the membership functions.

The procedure for generating *fuzzy relational models* is the same with some additions: each relation has its own weight factor and the number of membership functions for both the input and the output variables is about the same.

3 MODELLING

Fuzzy models were developed with the *FuzzEqu* Toolbox in *Matlab*® environment. Data was first filtered and after that membership definitions were generated. Then linguistic equations were developed from the data linearised with membership definitions. Linguistic rules were formed through linguistic relations. Finally, we are able to generate different types of fuzzy models (linguistic fuzzy models, singleton fuzzy models and Takagi-Sugeno models).

Fuzzy models and linguistic equations were compared. Different fuzzy models were constructed from linguistic equations using different shapes of membership functions and different number of membership functions. This approach was compared with the ANFIS method, which constructs fuzzy models directly from data.

Various alternative types of fuzzy models formed in many different ways were compared to see the effect on output. This comparison helps us to get better understanding for different fuzzy models. We can also compare fuzzy models with linguistic equations. Number of parameters used in models is important: the more parameters are used the better results we can get for the training data. But too many parameters result overfitting and models are not useable. Therefore, validation data is used for stopping the training. Finally, the resulting models are tested with an independent data set.

3.1 Measurements

For training, validation and testing, data from CLA 2000 measurements together with on-line Kappa-number measurements were used. The same data set was also used in earlier comparisons of modelling techniques (Murtovaara et al., 1999). The data contained 600 measurements for both training and validation, and 321 measurements for testing. The validation vectors are used to stop training early if further training on the primary vectors will hurt generalisation to the validation vectors.

Collecting the data has encountered several difficulties. Because of that there is not so much data available and test data differs slightly from other data. On the other hand, this material is good for testing the robustness of the models. If too many parameters are used, the model does not work in different operating conditions and therefore cannot be used in real applications.

3.2 Linguistic equations

In the kraft cooking case, automatic model generation provides a real valued interaction matrix

$$[-0.632\ -0.3435\ -1.6525\ -2.2145].$$

Effects of different variables can be clearly seen by dividing the matrix by its smallest value: the resulting matrix is

$$[1.84\quad 1\quad 4.81\quad 6.45].$$

Second variable (conductivity) has the smallest effect on Kappa-number. The first variable (refractive index) has 1.8 times bigger effect and third variable (temperature) 4.8 times bigger effect on kappa than conductivity. Fourth number in the matrix corresponds to the output (Kappa number). It has the biggest value because all three inputs have an effect on it.

This model is close the earlier LE model with integer coefficients (Murtovaara et al., 1999)

$$[-2\ \ 1\ -6\ -6].$$

Non-linearities are handled with membership definitions (Figure 2) which were generated for each variable from training data.

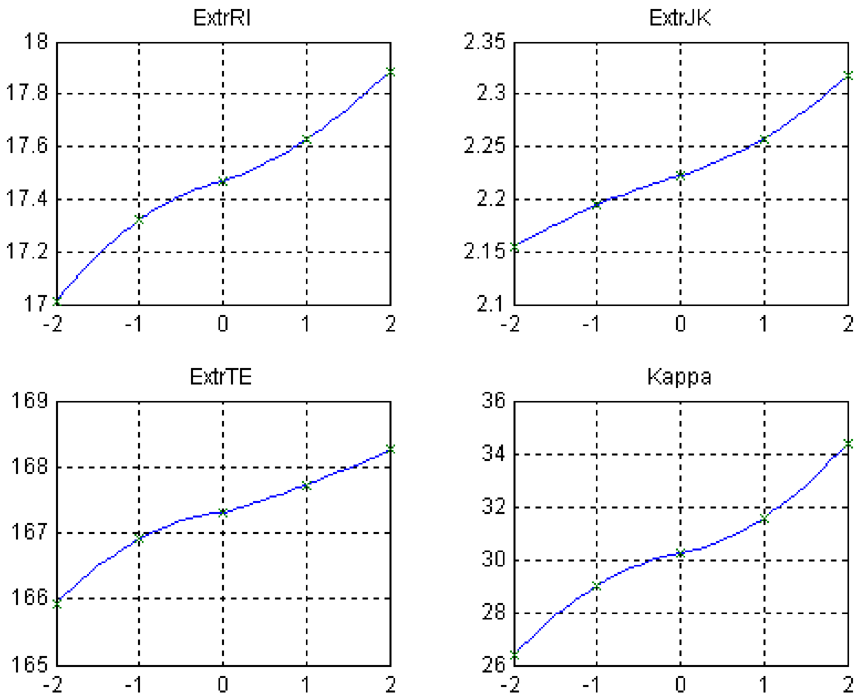

Figure 2. Membership definitions

Membership functions can be generated from membership definitions on any fuzzy partition. Figure 3 shows an example with three membership functions for each variable. Because of monotonously increasing membership definitions linguistic meanings of these membership functions are always in a correct sequence.

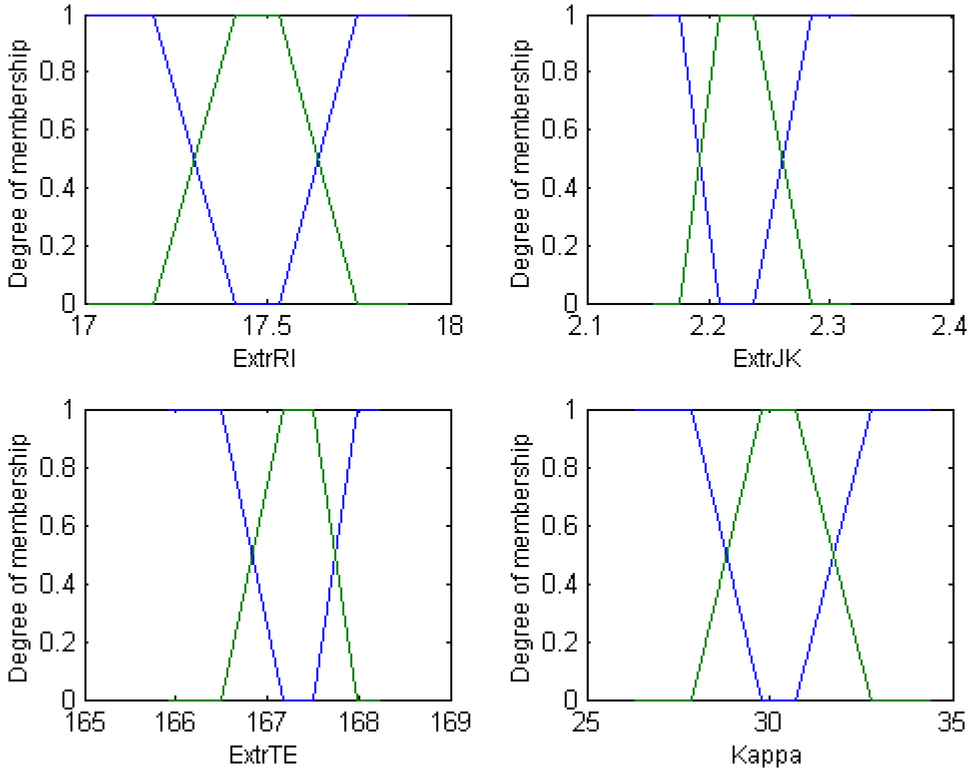

Figure 3. Membership functions

3.3 Linguistic relations

Linguistic relations can be generated directly from linguistic equations. Using five membership functions for each input we get 125 relations (5*5*5=125). Linguistic relations are actually only a ”substage” between linguistic equations and rules. Relations can be developed from the equations either using simultaneous method or sequential method. In models developed by simultaneous method, every relation consist all variables, and it is therefore more compact. Sequential method produces larger models with more relations, but it fits better for understanding and simplifying the rule base. In this case, we have only one linguistic equation. The number of membership functions could also be different for different variables.

Input and output variables must be selected before the development of relations. Locations of the membership functions are calculated with the *FuzzEqu Toolbox*. Since these locations are not necessarily identical with the locations defined for the output variable, the resulting linguistic relations are fuzzy. Relations are also fuzzy if the number of the calculated locations is higher than the number of membership functions defined for the output variable. Especially for real valued linguistic equations, the number of calculated locations is very high, e.g. 125 membership functions for the consequent variable would be much too high. The number of membership functions for the antecedent variable can also be limited.

Fuzzy relational models are based on a low number of membership functions for the antecedent variable. These systems can be tested as linguistic fuzzy models where same antecedents have several consequents with different weight factors.

3.4 Linguistic rules

Linguistic rules can be obtained from the relations (from 125 relations we will get 125 rules). Number of rules depends on the number of locations defined for the output variable, i.e. the starting system is a set of fuzzy relations. Different types of fuzzy models can be generated from these relations:

- For linguistic fuzzy models, there are membership functions for output too, and the consequent with highest degree of membership is chosen.
- For singleton fuzzy models, the consequent is a crisp number. The number of output singletons can be higher: all the rules may have a different singleton value.
- For Takagi-Sugeno fuzzy models, highly overlapping membership functions of antecedent variables will provide smoothly changing models. Also the number of membership functions and rules should be fairly low. The *FuzzEqu Toolbox* generates local linear models from the LE model.
- Fuzzy relational models are handled as linguistic fuzzy models. The difference is that each antecedent combination can have several dissimilar consequents. The weight factors of these rules are defined from the degree of membership obtained for the corresponding relation. The number of active rules depends on the chosen threshold of membership.

All these models have been tested with the *Matlab Fuzzy Logic Toolbox.*

The models developed with these approaches have been compared to the neuro-fuzzy *ANFIS* method available in the *Fuzzy Logic Toolbox.* The initial structure of the systems for the ANFIS tuning were generated with default grid partitioning technique. The number of membership functions can be chosen independently for each input variable. The number of different consequent surfaces is the same as the number of rules generated.

4 RESULTS

The modelling results are shown graphically: three figures on model fitting (one for each data type) plus a model surface. All these should be compared when selecting the best model. However, we cannot show all these details because we have constructed very many models. The model comparison is in this report done on the basis of the numerical fitting results. It is a pity because figures tell us the true behaviour of models – numerical values can be good even if models do not work properly.

4.1 Linguistic equation models

As a starting point of generating fuzzy models is the linguistic equation model. The real valued model has a better performance both in training and in testing (Table 1). Correlation is slightly improved. Root mean square error (RMS) reduces considerably to around 1.0. Relative error reduces to 3 percent. As the LE model was not tuned at all, validation can be considered as another testing.

Tuning of the real valued linguistic equation (RLE) model improved correlation and RMS for the training data. For validation data, both relative error and RMS were improved but correlation became slightly smaller. For testing data, performance was slightly worse.

Table 1. Performance of linguistic equation models

	Correlation	Relative error	RMS
Integer valued coefficients			
*training	0.75	0.045	1.37
*validation	0.64	0.039	1.15
*testing	0.74	0.051	1.56
Real valued coefficients			
*training	0.77	0.033	1.01
*validation	0.67	0.033	0.96
*testing	0.70	0.032	0.98
RLE model after tuning			
*training	0.78	0.033	0.99
*validation	0.65	0.028	0.82
*testing	0.69	0.033	1.00

Prediction results of the real valued linguistic equations (without tuning) are shown in Figures 4, 5 and 6. As the range of the validation data is quite narrow, validation stop is not achieved and the improvement of the fitting result continues all the time in tuning.

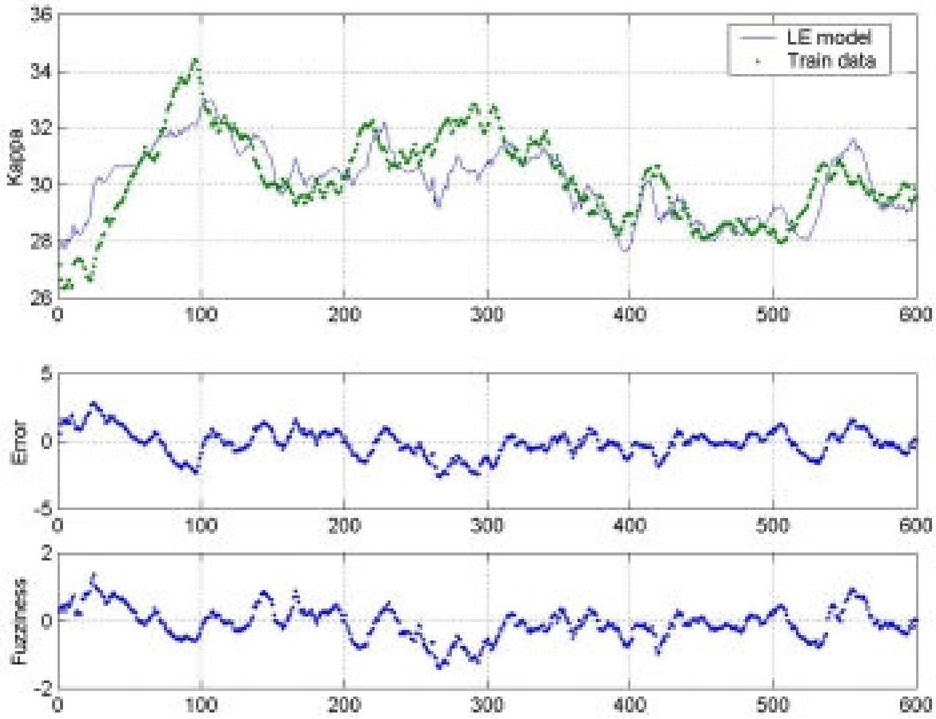

Figure 4. Kappa prediction with LE model for train data.

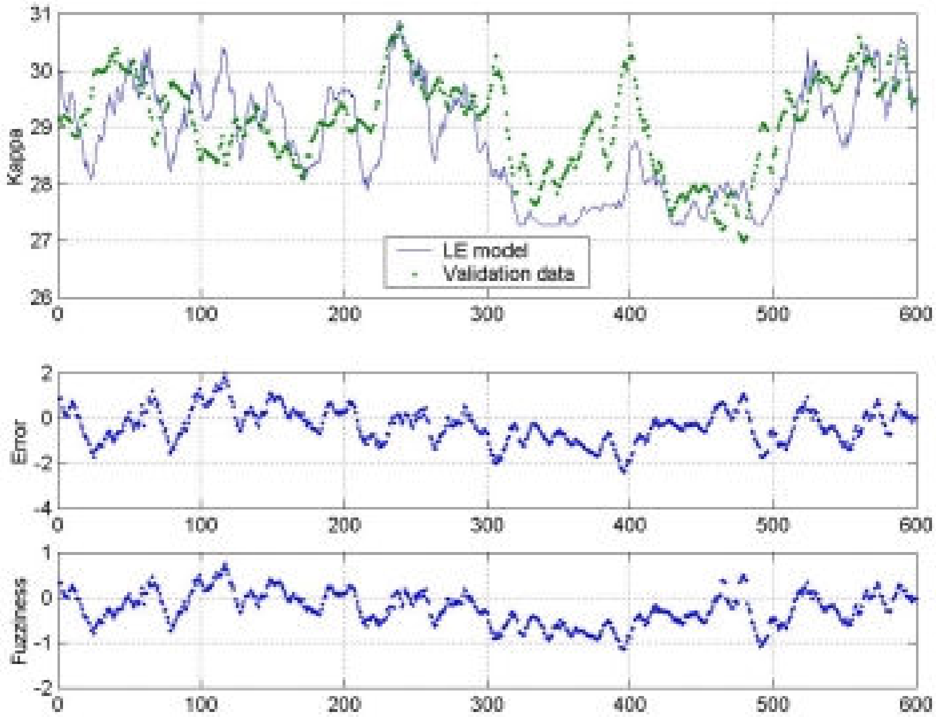

Figure 5. Kappa prediction with LE model for validation data.

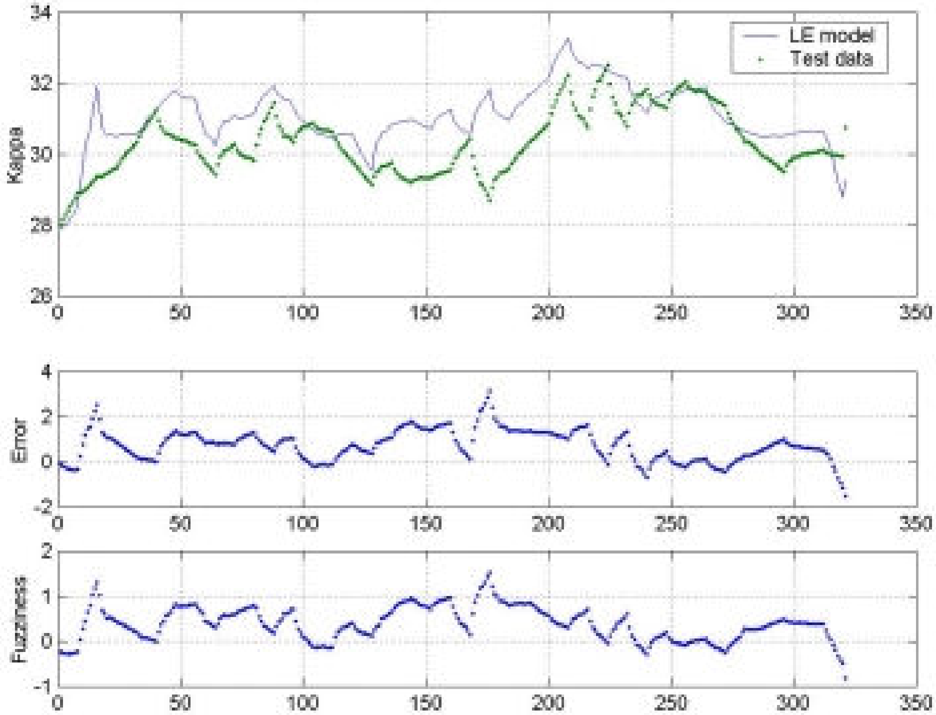

Figure 6. Kappa prediction with LE model for test data.

4.2 Fuzzy modelling with linguistic equations

Fuzzy models were constructed from different types of linguistic equations. Both linear and non-linear membership functions were used. Models based on different number of rules were also compared.

4.2.1 Linear membership functions

Results for models developed from real valued linguistic equations using triangular membership functions are shown in Table 2 and Figure 7, 8 and 9. For Takagi-Sugeno type model, the number of rules is too high. Although fitting to the train data is the best of the alternatives shown, fitting to the test data is the worst.

Real valued linguistic equations were better than integer valued equations (Table 1). This difference can be seen also in corresponding fuzzy models: results the fuzzy models based on integer valued equations (Table 3) are not as good as the results shown in Table 2.

Table 2. Real valued coefficients, 125 rules, five linear triangular membership functions for each variable.

	Correlation	Relative error	RMS
Linguistic fuzzy model			
*training	0.73	0.037	1.11
*validation	0.58	0.029	0.84
*testing	0.57	0.030	0.98
Singleton fuzzy model			
*training	0.76	0.034	1.03
*validation	0.63	0.028	0.81
*testing	0.61	0.035	1.07
Takagi-Sugeno fuzzy model			
*training	0.78	0.033	1.00
*validation	0.66	0.029	0.85
*testing	0.72	0.040	1.17

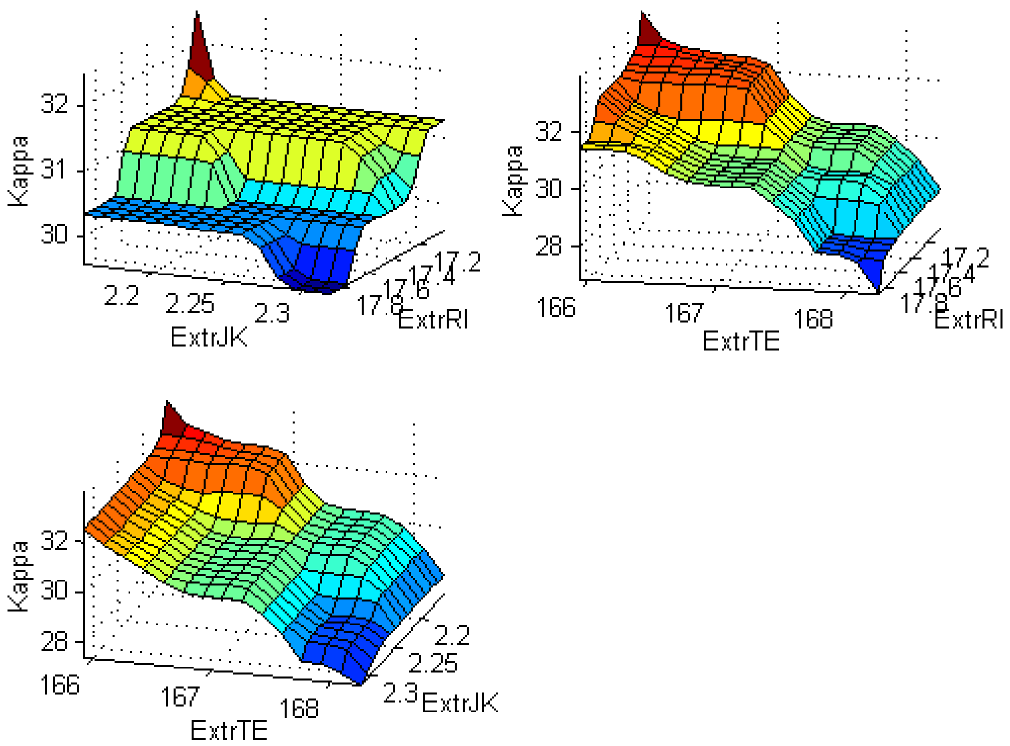

Figure 7. Linguistic fuzzy model surface.

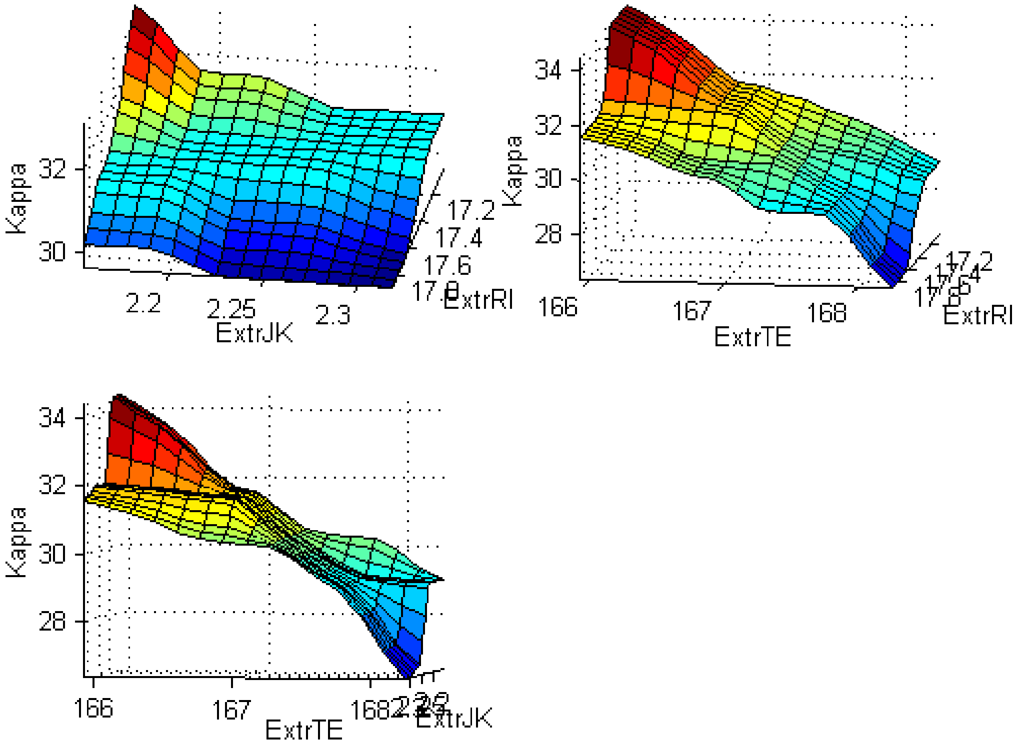

Figure 8. Singleton fuzzy model surface

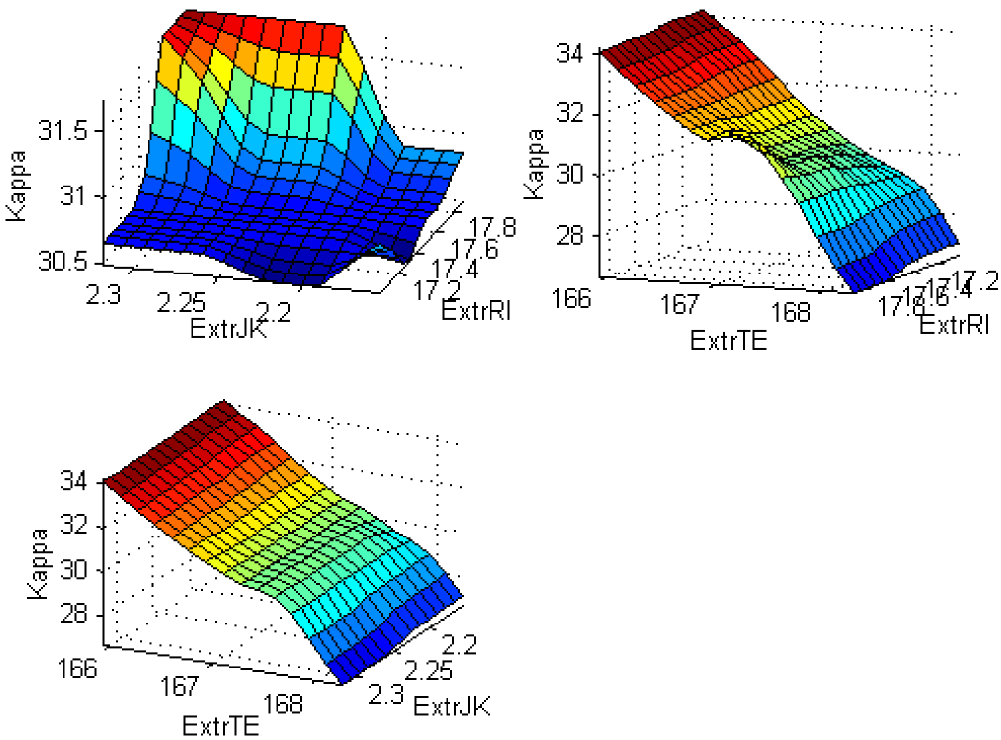

Figure 9. Takagi-Sugeno -type model surface

Table 3. Integer valued coefficients, 125 rules, five linear triangular membership functions for each variable.

	Correlation	Relative error	RMS
Linguistic fuzzy model			
*training	0.75	0.043	1.32
*validation	0.60	0.042	1.22
*testing	0.65	0.064	1.93
Singleton fuzzy model			
*training	0.75	0.037	1.12
*validation	0.61	0.034	0.98
*testing	0.64	0.050	1.52
Takagi-Sugeno fuzzy model			
*training	0.75	0.034	1.04
*validation	0.60	0.026	0.75
*testing	0.73	0.035	1.07

Table 4. Real valued coefficients, 125 rules, five linear trapezoidal membership functions for each variable.

	Correlation	Relative error	RMS
Linguistic fuzzy model			
*training	0.70	0.038	1.14
*validation	0.56	0.032	0.96
*testing	0.55	0.039	1.20
Singleton fuzzy model			
*training	0.73	0.039	1.19
*validation	0.56	0.041	1.18
*testing	0.57	0.050	1.52
Takagi-Sugeno fuzzy model			
*training	0.77	0.033	1.01
*validation	0.66	0.032	0.94
*testing	0.71	0.042	1.30

While using trapezoidal membership functions, results are worse, especially for singleton models (Table 4). For Takagi-Sugeno type models, the effect is smallest. This is quite natural since membership functions with wide core areas should be preferable for TS models.

4.2.2 Non-linear membership functions

Converting the shape of membership functions from linear to non-linear was done for triangular membership functions (Table 5). Difference to corresponding linear systems (Table 2) is very small.

Table 5. Real valued coefficients, 125 rules, five non-linear triangular membership functions for each variable.

	Correlation	Relative error	RMS
Linguistic fuzzy models			
*training	0.71	0.037	1.13
*validation	0.58	0.030	0.86
*testing	0.56	0.034	1.01
Singleton fuzzy models			
*training	0.75	0.035	1.05
*validation	0.62	0.022	0.82
*testing	0.58	0.036	1.10
Takagi-Sugeno fuzzy models			
*training	0.78	0.033	1.00
*validation	0.66	0.029	0.85
*testing	0.72	0.040	1.17

4.2.3 Number of membership functions

Results of models with different number of membership functions are shown in Tables 6, 7 and 8. Each variable can have different number of membership functions. For example membership function [2 2 3 3] means that two membership functions have been used for first and second variable and three functions for third and fourth (output). Also different shapes of membership functions were compared.

Table 6 shows the results for Takagi-Sugeno type models. Triangular membership functions are again slightly better than the trapezoidal ones. Differences between linear and non-linear versions are also very small. Testing results are best for the cases with 12 rules.

Table 6. Takagi-Sugeno fuzzy model using different number of membership functions.

Rules	Membership functions	Linear/ non-linear	Shape	Data	Correlation	Relative error	RMS
12	2 2 3 3	linear	trapezoidal	*training	0.77	0.033	1.01
				*validation	0.66	0.026	0.75
				*testing	0.72	0.042	1.26
12	2 2 3 3	non-linear	trapezoidal	*training	0.77	0.033	1.01
				*validation	0.66	0.026	0.75
				*testing	0.72	0.041	1.26
12	2 2 3 3	linear	triangular	*training	0.78	0.033	0.98
				*validation	0.65	0.027	0.79
				*testing	0.72	0.038	1.15
12	2 2 3 3	non-linear	triangular	*training	0.79	0.033	0.98
				*validation	0.65	0.027	0.80
				*testing	0.72	0.038	1.15
8	2 2 2 2	linear	trapezoidal	*training	0.79	0.033	1.01
				*validation	0.67	0.025	0.73
				*testing	0.71	0.043	1.29
16	2 2 4 4	linear	trapezoidal	*training	0.77	0.035	1.05
				*validation	0.67	0.030	0.88
				*testing	0.72	0.044	1.33
125	5 5 5 5	linear	trapezoidal	*training	0.77	0.033	1.01
				*validation	0.66	0.032	0.94
				*testing	0.71	0.042	1.30

Table 7. Linguistic fuzzy model using different number of membership functions.

Number of membership functions	Data	Correlation	Relative error	RMS
7	*training	0.77	0.034	1.02
	*validation	0.62	0.027	0.77
	*testing	0.61	0.042	1.27
6	*training	0.75	0.035	1.06
	*validation	0.63	0.026	0.74
	*testing	0.64	0.042	1.27
5	*training	0.73	0.037	1.11
	*validation	0.58	0.029	0.84
	*testing	0.57	0.030	0.98
4	*training	0.74	0.035	1.07
	*validation	0.54	0.025	0.73
	*testing	0.56	0.038	1.16
3	*training	0.70	0.040	1.21
	*validation	0.55	0.035	1.03
	*testing	0.71	0.024	0.72

Table 8. The effect of Singleton values.

Number of Singletons	Data	Correlation	Relative error	RMS
5	*training	0.76	0.035	1.05
	*validation	0.59	0.036	1.05
	*testing	0.62	0.041	1.24
9	*training	0.77	0.034	1.02
	*validation	0.65	0.028	0.81
	*testing	0.72	0.034	1.04
11	*training	0.77	0.034	1.02
	*validation	0.67	0.028	0.83
	*testing	0.74	0.033	1.00
23	*training	0.78	0.033	1.01
	*validation	0.65	0.029	0.85
	*testing	0.72	0.034	1.03

For linguistic fuzzy models, large number of membership functions may result overfitting (Table 7). For singleton fuzzy models, a large number of singletons is beneficial (Table 8) but the improvement becomes insignificant with further increase of singletons.

4.3 Fuzzy relational models

Fuzzy relational models were generated from linguistic equation models. The number of rules is defined by the threshold. Each relation, whose degree of membership reaches the threshold, is included. The number of rules in the corresponding linguistic fuzzy system decreases with increasing threshold value (Table 9). The performance of the resulting system becomes slightly worse when more rules are removed. Overfitting can be seen in testing results.

Table 9. Linguistic fuzzy model. Five membership functions using fuzzy relations, different number of rules.

	Correlation	Relative error	RMS
241 rules, threshold=0			
*training	0.75	0.034	1.05
*validation	0.64	0.024	0.70
*testing	0.62	0.043	1.32
195 rules, threshold=0.2			
*training	0.75	0.035	1.06
*validation	0.63	0.024	0.71
*testing	0.62	0.044	1.34
149 rules, threshold=0.4			
*training	0.76	0.035	1.05
*validation	0.64	0.025	0.72
*testing	0.61	0.045	1.36
101 rules, threshold=0.6			
*training	0.77	0.035	1.04
*validation	0.60	0.028	0.81
*testing	0.55	0.045	1.35
65 rules, threshold=0.75			
*training	0.75	0.036	1.09
*validation	0.54	0.031	0.90
*testing	0.61	0.045	1.36
41 rules, threshold=0.85			
*training	0.72	0.038	1.15
*validation	0.56	0.033	0.95
*testing	0.59	0.046	1.40

Locations of the membership functions are very important: here they were defined beforehand for all variables. In addition to this we can compare two models using same number of rules (Table 10). The result is considerably better when only four membership functions are for each variable but all the rules are accepted.

Table 10. Performance of fuzzy set systems based on different methods using the same amount of rules.

65 rules, fuzzy relations, threshold=0.75, five membership functions each				
*training	0.75	0.036	1.09	
*validation	0.54	0.031	0.9	
*testing	0.61	0.045	1.36	
64 rules, fuzzy linguistic model, four membership functions each				
*training	0.76	0.034	1.04	
*validation	0.55	0.026	0.76	
*testing	0.62	0.039	1.19	

4.4 ANFIS modelling

We can have fairly good results by using ANFIS (adaptive neuro-fuzzy inference system). ANFIS constructs Takagi-Sugeno type fuzzy models directly from data. The results with two or three membership functions are very similar but when using four, we can notice some signs of overfitting. Figures 10, 11 and 12 show the behaviour of an ANFIS model using three membership functions for each variable. Figure 13 is the corresponding model surface.

Table 11. Performance of ANFIS models with different number of membership functions used.

Number of Membership functions	Data	Correlation	Relative Error	RMS
2 2 2 2	*training	0.81	0.031	0.94
	*validation	0.69	0.020	0.59
	*testing	0.71	0.036	1.10
3 3 3 3	*training	0.82	0.030	0.92
	*validation	0.71	0.020	0.59
	*testing	0.69	0.035	1.08
4 4 4 4	*training	0.84	0.029	0.87
	*validation	0.71	0.020	0.59
	*testing	0.65	0.044	1.32

Both in training and in validation, the fitting results are better than in the case of linguistic equations (Table 1). However, test results with an independent data set are worse.

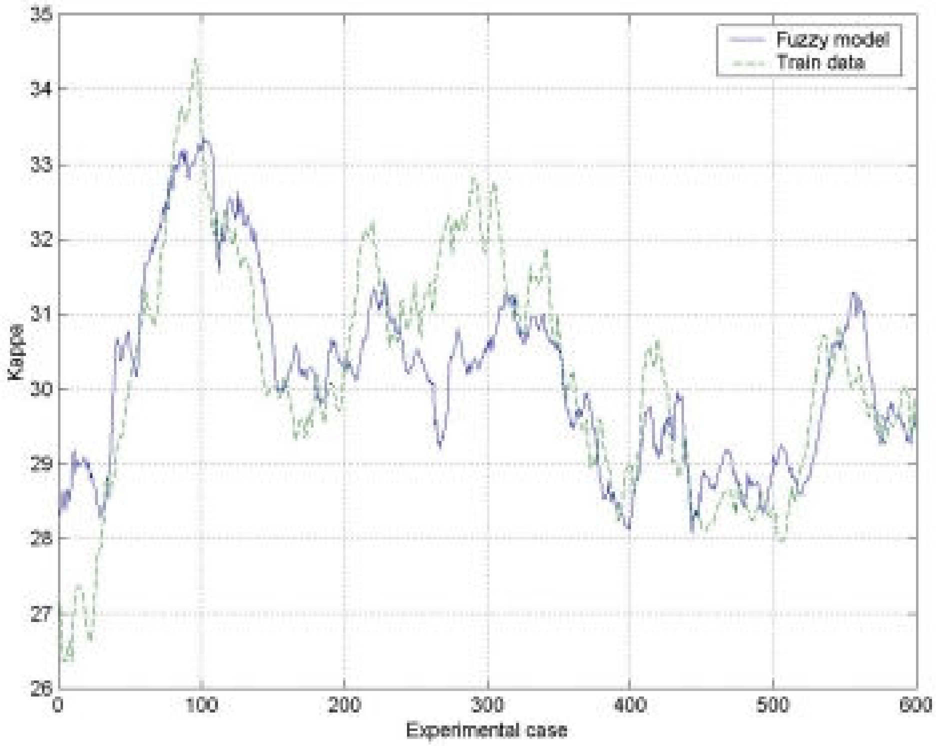

Figure 10. Comparing training data and ANFIS model.

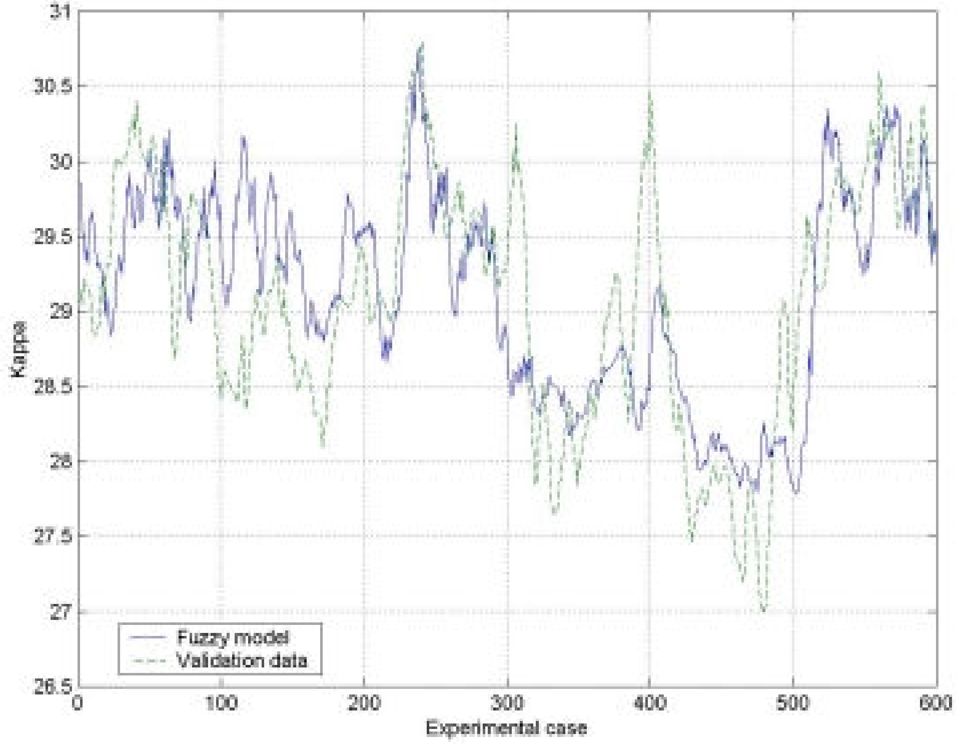

Figure 11. Comparing validation data and ANFIS model.

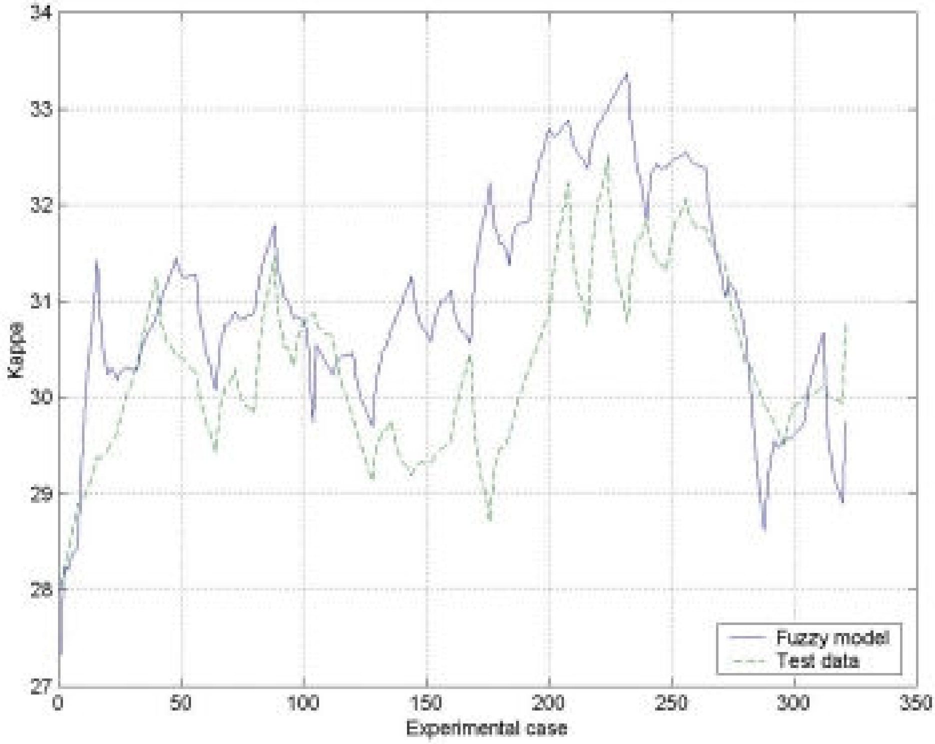

Figure 12. Comparing testing data and ANFIS model.

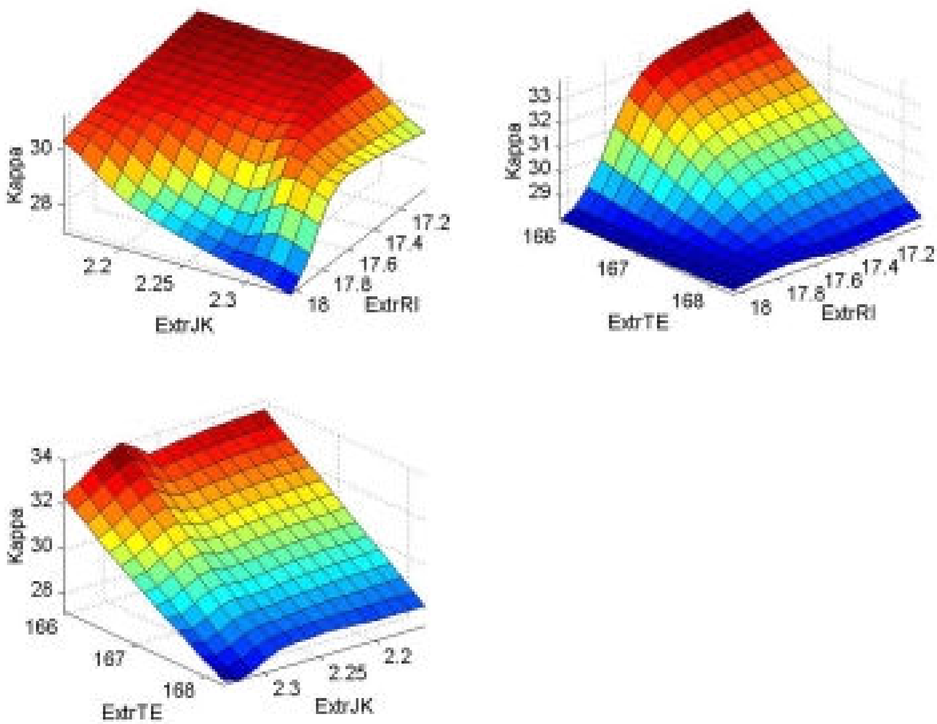

Figure 13. Model surface for ANFIS model using three membership functions.

5 DISCUSSION

5.1 Number of parameters

The number of parameters used in a model is important for goodness and adaptiviness of the fuzzy model. For example if too few parameters are used, model neglects the non-linearities of process and we have to cope with linear effect which does not work so well. On the contrary if parameters are used too many (for example seven membership functions for each variable) it may happen so called overfitting. It means that model can remember the learning data. Then model works almost perfect for training data, but it does not work with different data (for example test data). When the number of parameters is suitable, the model works quite well for both train and test data. This kind of model can be used in industrial applications too, because it works in different conditions and therefore it is robust. Figures 14, 15 and 16 show the effect of overfitting. For train data the model works very well (Figure 14), but already with validation data there are problems (Figure 15). Same problems continue with test data.

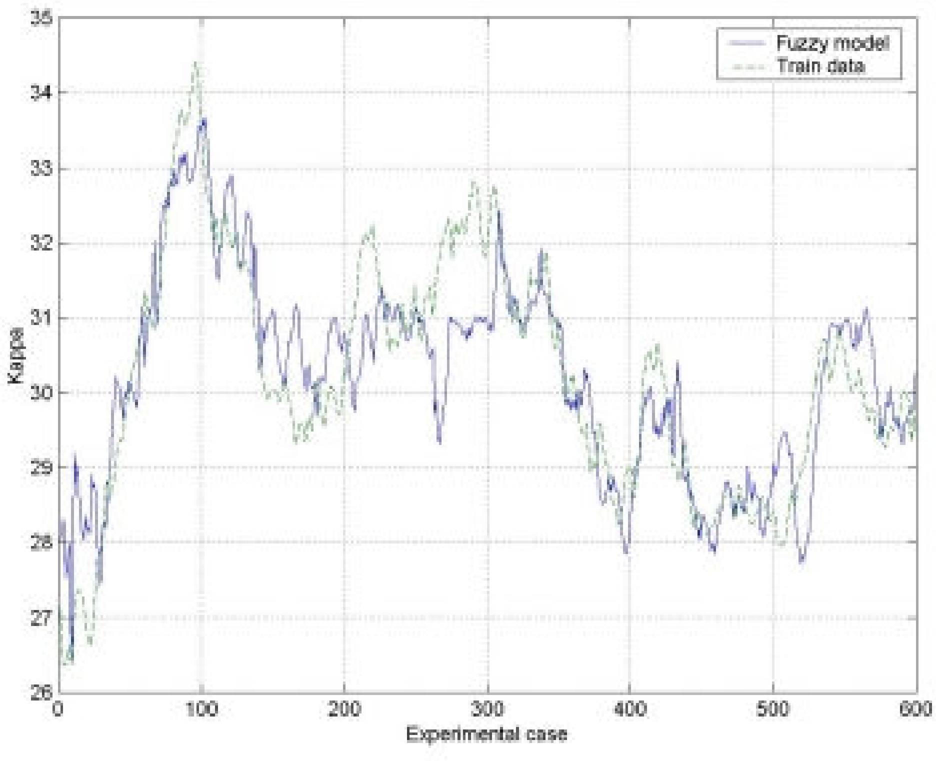

Figure 14. Model for train data. (overfitting)

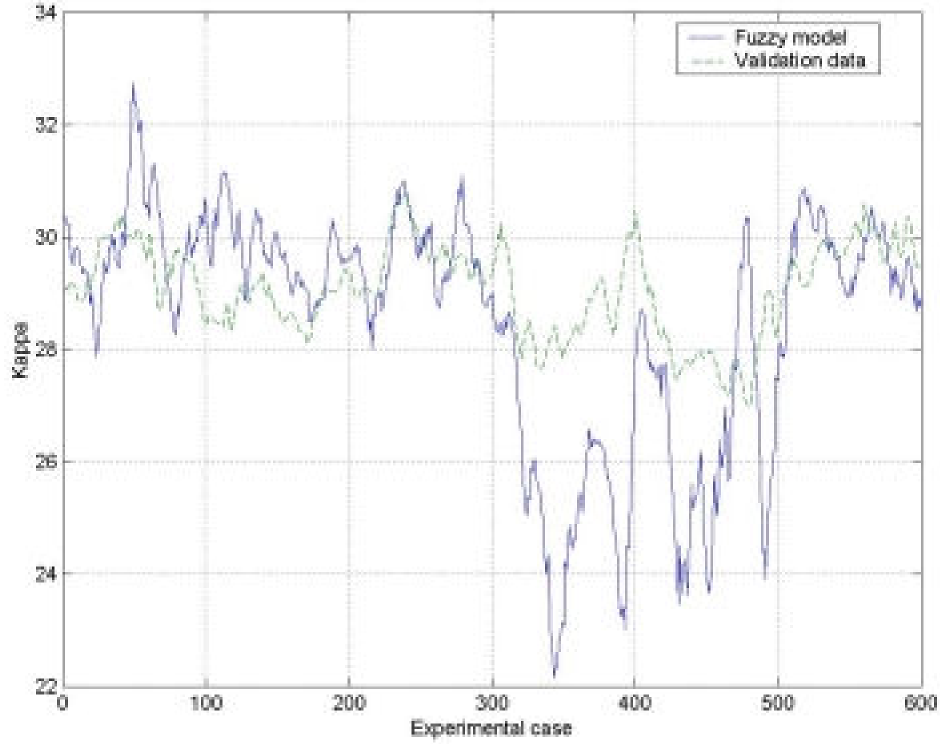

Figure 15. Model for validation data. (overfitting)

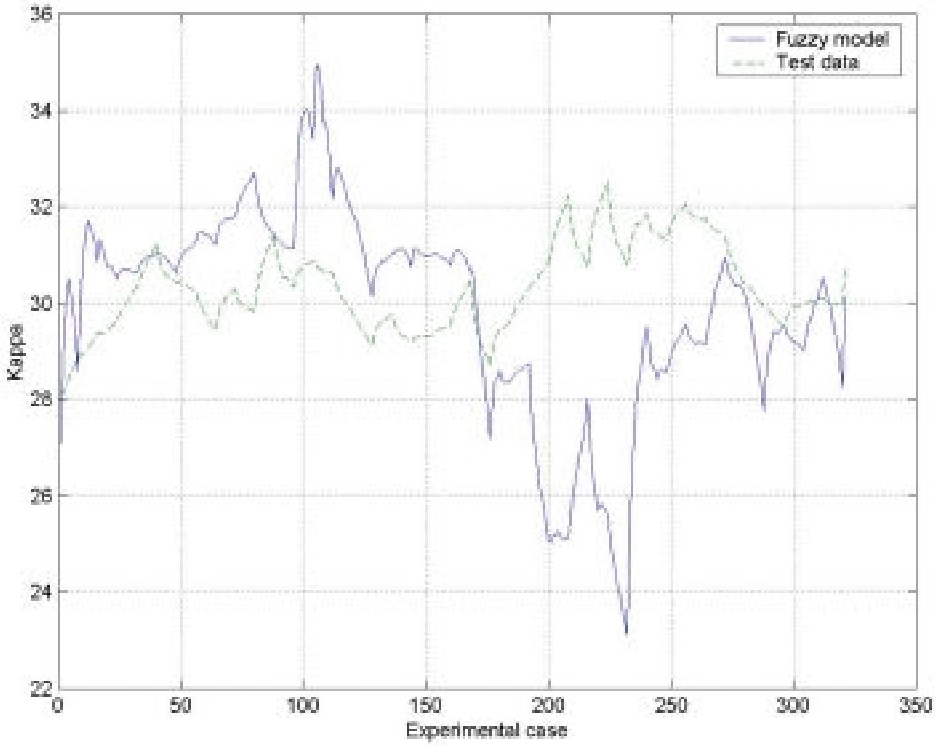

Figure 16. Model for test data. (overfitting)

5.2 Model performance

Correlation, relative error and square error tell us quite a lot about the performance of a model. Model is, of course, good when correlation is high and errors are as small as possible. But we cannot rely only on these numerical values – studying the fitting in detail gives us a lot of additional information. It may be that numerical values are good, but fitting results of the model are not good in some operating area. For example Figure 17 shows that operating is not good though numerical values are good. Figures 18, 19 and 20 present the different models for the same data. Numerical values are almost same (Table 2) but fitting results show that there are differences in operation.

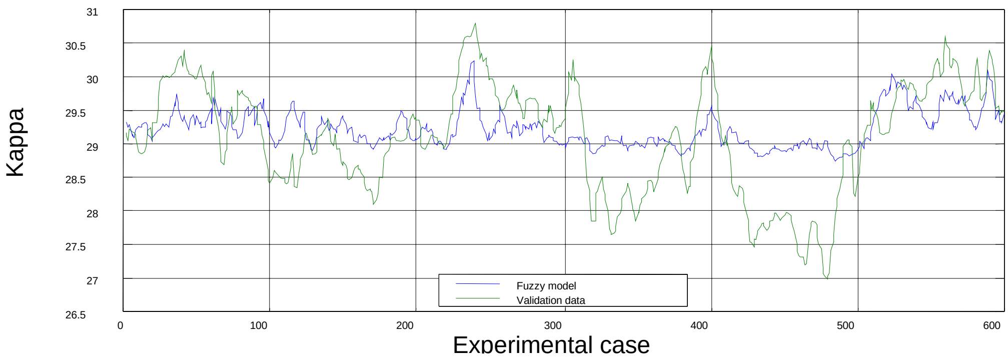

Figure 17. Numerical values are good: Correlation: 0.66, relative error: 0.023, RMS: 0.68, but figure shows that model is not good at all.

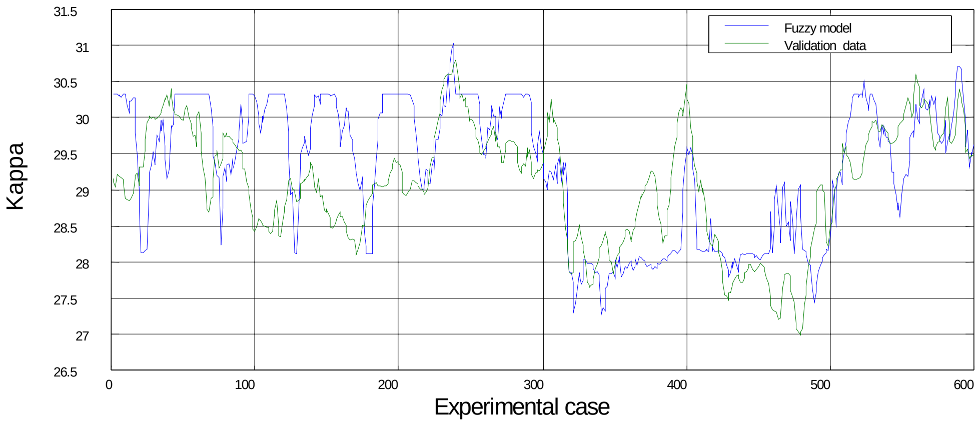

Figure 18. Linguistic fuzzy model.

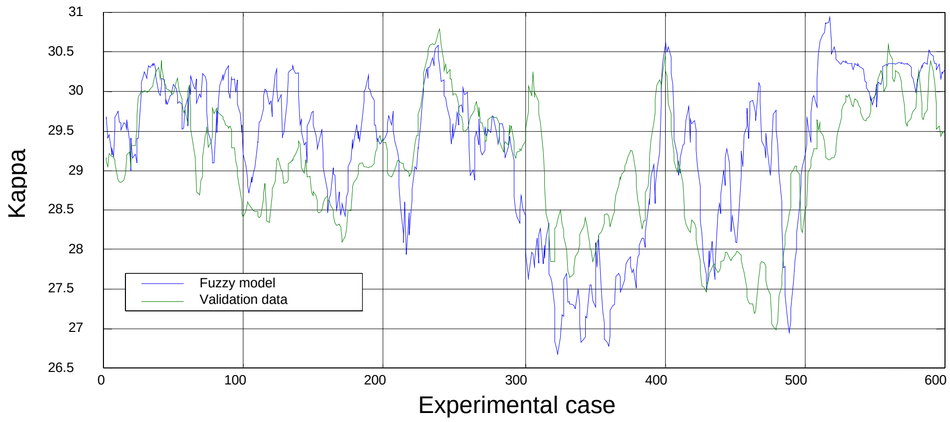

Figure 19. Takagi-Sugeno type fuzzy model.

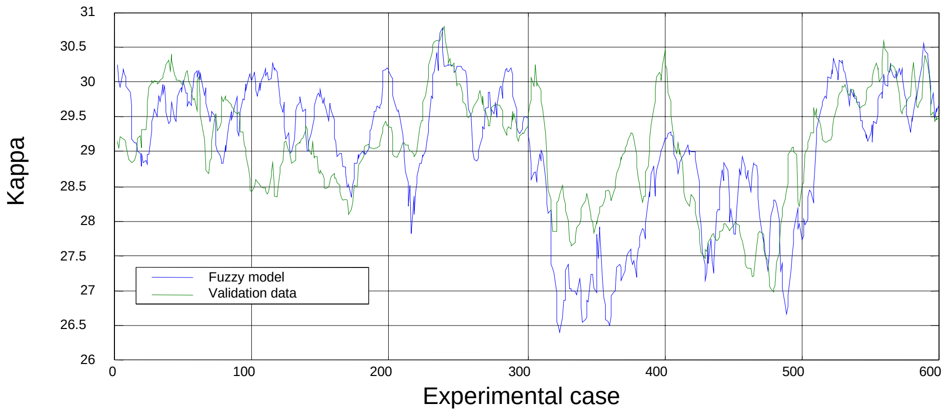

Figure 20. Singleton fuzzy model.

6 CONCLUSIONS

The results given by linguistic equations are the reference for fuzzy models. For fuzzy models, altering the shape of membership function has an effect on results. Triangular membership functions give better results with linguistic fuzzy models and singleton fuzzy models. Takagi-Sugeno models gave also a little better results when triangular membership functions are used, usually trapezoidal membership functions are better for TS models. Increasing singleton values gives better results. For Takagi-Sugeno models we can get better results using less membership functions. Linguistic fuzzy models and singleton fuzzy models give the better results the more membership functions are used.

We also compared the effect using different number of rules. Fuzzy relations were used and by defining the threshold, different number of rules was included. Of course decreasing number of rules gives a little bit worse results. If the number of rules is same we can note that normal relations work better though there are less membership functions used.

Linguistic equations (LE) work well for this data. For the test data, the performance of the real-valued LE model was the best althougt a better fitting accuracy with training data was obtained by constructing Takagi-Sugeno (TS) fuzzy models with the ANFIS method. There are also overfitting problems with the TS models. TS models could be improved by using subtractive clustering in the generation of the initial structure. This approach results less rules which is beneficial for TS systems.

Easy configuration and robustness are the main benefits of the LE models. The fitting performance must be compared to the number of modelling parameters. Since only five parameters are needed for each variable, the LE system can, more easily, be adapted to various operating conditions. This means big advantage compared to all the fuzzy systems tested in this study.

REFERENCES

Babuska, R. (1997). Fuzzy Modeling and Identification.Delft University of Technology, Delft, The Netherlands.

Driankov, D., Hellendoorn, H., and Reinfrank, M. (1993). An Introduction to Fuzzy Control. Springer, Berlin, Germany.

Haataja, K., Leiviskä, K. and Sutinen, R. (1997). Kappa-number estimation with neural networks. In: Proceedings of IMEKO World Congress, Finnish Society of Automation, Tampere, Finland, Volume XA, pp. 1–5.

Jang, J.-S. R. (1993). ANFIS: Adaptive-network-based fuzzy inference system, IEEE Transactions on Systems, Man and Cybernetics, 23 (3), 665-685.

Juuso, E. K. (1999a). Fuzzy Control in Process Industry: The Linguistic Equation Approach. In: *Fuzzy Algorithms for Control*, (Verbruggen H. B., Zimmermann H.-J. And Babuska R. ed.), International Series in Intelligent Technologies, Kluwer, Boston, 1999, pp. 243 - 300.

Juuso, E. K., (1999b). Intelligent Systems Design with Linguistic Equations. *9th Workshop Fuzzy Control* des GMA-FA 5.22 am 4/5.11.1999, Dortmund, Deutschland. Forschungs-bericht Nr. 0449, Universität Dortmund, Fakultät für Electrotecknik, Dortmund, 1999, pp. 177 - 196.

Juuso, E. K., Ahola, T., and Leiviskä, K., (1997). Fuzzy Modelling of a Rotary lime Kiln Using Linguistic Equations and Neuro-fuzzy Methods. 3rd IFAC Symposium of Intelligent Components and Instruments for Control Applications – SILICA ’97, Annecy, France, June 9-11, 1997, pp. 579-584.

Krone, A. and Kiendl, H. (1994). Automatic generation of positive and negative rules for two-way fuzzy controllers. In Zimmermann, H.-J., editor, Proceedings of the Second European Congress on Intelligent Technologies and Soft Computing – EUFIT'94, Aachen, September 21 - 23, 1994, volume 1, pp. 438-447, Aachen. Augustinus Buchhandlung.

Krone, A. and Schwane, U. (1996). Generating fuzzy rules from contradictory data of different control strategies and control performances. In Proceedings of the Fuzz-IEEE'96, New Orleans, USA, pp. 492-497.

Murtovaara S., Leiviskä K., Juuso E., Sutinen R., (1999). Modelling of Pulp Characteristics in Kraft Cooking. Report A No 9, December 1999, Control Engineering Laboratory, University of Oulu, Oulu, 1999, 20p.

Pedrycz W. (1984). An Identificaton Algorithm in Fuzzy Relational Systems, Fuzzy Sets and Systems, 13, 153-167.

Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modelling and control. IEEE Trans. Syst., Man, & Cybern. 15(1):116-132.

Wang, L.-X. (1994). Adaptive Fuzzy Systems and Control. Design and stability Analysis. Prentice Hall, New Jersey.

Wang, L.-X. and Mendel, J. (1992). Generating fuzzy rules by learning from examples. IEEE Transactions on Systems, Man, and Cybernetics 22:1414--1427.

ACKNOWLEDGEMENT

The development of the Quality Forecasting Tool was funded by TEKES and ABB Industry Oy as a part of TOOLMET-2-MODIPRO project in technology programme *Adaptive and Intelligent Systems Applications*.

ISBN 951-42-5546-1
ISSN 1238-9390
University of Oulu
Control Engineering Laboratory

1. Yliniemi L., Alaimo L., Koskinen J., Development and Tuning of A Fuzzy Controller for A Rotary Dryer. December 1995. ISBN 951-42-4324-2.
2. Leiviskä K., Simulation in Pulp and Paper Industry. February 1996. ISBN 951-42-4374-9.
3. Yliniemi L., Lindfors J., Leiviskä K., Transfer of Hypermedia Material through Computer Networks. May 1996. ISBN 951-42-4394-3.
4. Yliniemi L., Juuso E., (editors), Proceedings of TOOLMET'96- Tool Environments and Development Methods for Intelligent Systems. May 1996. ISBN 951-42-4397-8.
5. Lemmetti A., Leiviskä K., Sutinen R., Kappa number prediction based on cooking liquor measurements. May 1998. ISBN 951-42-4964-X.
6. Jaako J., Aspects of process modelling. September 1998. ISBN 951-42-5035-4.
7. Lemmetti A., Murtovaara S., Leiviskä K., Sutinen R., Cooking variables affecting the kraft pulp properties. June 1999. ISBN 951-42-5309-4.
8. Donnini P.A., Linguistic equations and their hardware realisation in image analysis. June 1999. ISBN 951-42-5314-0.
9. Murtovaara S., Juuso E., Sutinen R., Leiviskä K., Modelling of pulp characteristics in kraft cooking. December 1999. ISBN 951-42-5480-5.
10. Cammarata L., Yliniemi L., Development of a self-tuning fuzzy logic controller (STFLC) for a rotary dryer. December 1999. ISBN 951-42-5493-7.
11. Isokangas A., Juuso E., Fuzzy Modelling with Linguistic Equations. February 2000. ISBN 951-42-5546-1.