Intelligent Dynamic Simulation of Batch Cooking

Esko K. Juuso
Control Engineering Laboratory
Department of Process and Environmental Engineering
P.O.Box 4300, 90014 University of Oulu, Finland
E-mail: esko.juuso@oulu.fi

Abstract

Intelligent dynamic simulator has been developed with linguistic equations *(LE)* for a *SuperBatch* coking process. The LE models consist of two parts: interactions are handled with linear equations, and nonlinearities are taken into account by scaling with membership definitions. Data-driven modelling was done by *FuzzEqu* toolbox on the basis of the measurements of the cooking liquor analyser *CLA 2000*: the concentrations of alkali, total dissolved solids and lignin are measured on-line during individual cooks in a batch digester house. According to extensive on-line tests in an industrial pulp mill, dynamic *LE* models are well suited for forecasting the cooking result: residual alkali, lignin and dissolved solids. The models are adapted to the changing operating conditions with configurable parameters.

1 Introduction

Cooking is the major process in the pulp mill and its proper control is very important to the pulp production. The aim of chemical pulping is to remove enough lignin from wood so that the fibers are free and give them the required characters at the lowest possible cost. Variation in the quality parameters is minimised by providing the similar cooking history for all chip particles. Chip quality variations, measurement problems and long process delays make the control difficult. The main problem is that the quality variables cannot be measured while the cooking goes on. Pulp cooking has used computerised control since the early 1960s [1].

Cooking reactions start when chips reach the cooking temperature, about 150 - 170 ^oC, depending on the wood species and grade requirements. The active chemicals of the cooking liquor react with lignin in chips and convert it chemically into the compounds that dissolve in the cooking liquor. Fibers are separated into the mass since the bonding material of the chips is dissolved. Cooking chemicals also react with fibrous material (cellulose and hemicellulose) and break down their molecular bonds. These reactions are to be avoided as much as possible because they lead to losses in the fiber yield and decrease the pulp strength.

There are two main types of process equipment for cooking: batch and continuous digesters. This paper deals with the batch cooking process, more specifically *SuperBatch* cooking, where the chips are pre-impregnated and preheated with warm black liquor (Figure 1) to make batch cooking more energy efficient.

The cooking liquor is circulated from the middle to top and bottom to ensure a uniform cooking. When the target H-factor has been reached, the circulation is stopped and the cool displacement liquor is pumped into the bottom part of the digester. The delignification process stops when the temperature decreases under 100 ^oC. The used cooking liquor is taken out from the top and directed to the accumulators to be reused.

Cooking liquor analyser CLA 2000 is an advanced measurement device developed for analysing the chemical pulping processes (Figure 1). It utilises a continuous sampling technique that means also, in practice, a continuous measurement if the analyser is applied for one sample point, only. In batch cooking, the cooking liquor analyser enables the analysis of alkali concentration and concentrations of total dissolved solids and dissolved lignin during individual batches. As the cooking progresses, all three measurements give information showing how cooking reactions go on. With batch digester

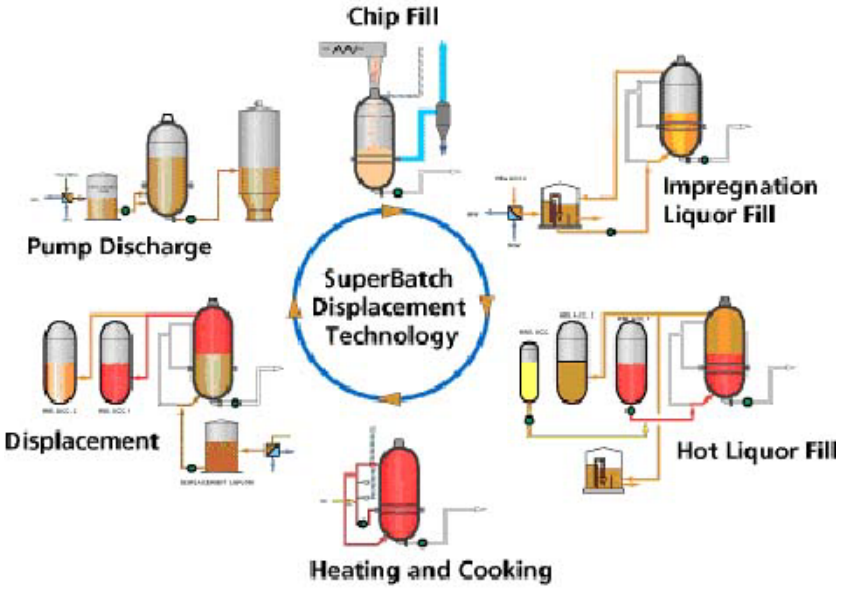

Figure 1: *SuperBatch* sequences and cooking liquor analyser *CLA 2000*.

operation it is possible to terminate the cooking process just when the required cooking degree has been achieved. Measurements are carried out in the actual process concentration and no dilution is needed. Measurements are indirect: alkali components are based on conductivity, solid contents on refractive index and dissolved lignin on UV-absorbency [2].

The widely used quality variable, the *Kappa number*, can be predicted on the basis of these measurements before the end of the cook. As the digester process is far from linear and simple input-output system, the analysis must be non-linear. Different approaches have been used for mathematical modelling of the cooking result [3]: fuzzy logic, partial least squares method (*PLS*), artificial neural networks (*ANN*) and linguistic equations (*LE*). In modelling the Kappa number in a continuous digester, all these methods seem to learn the process behaviour in a similar manner, but the *LE* models are the best in process environment since they can be adapted to various operating conditions in an understandable way. *ANN* and *PLS* models are sensitive for changes in process conditions, and fuzzy models need a large number of membership functions and rules that are too time-consuming to adapt.

In batch cooking, dynamic modelling and simulation is necessary. Dynamic *LE* models have been used in tuning *LE* controllers since they have provided accurate prediction and good performace in continuous processes, e.g. a lime kiln and a solar collector field [4]. For batch cooking, the dynamic *LE* modelling was started in 2000 [5]. According to the on-line tests, dynamic *LE* models are well suited for forecasting the cooking result. The approach is generic for batch processes: in [6] modelling of a Super Batch cooking process was compared to the modelling of a fluidised bed granulator, and later the approach has been applied in fed-batch fermentation [7].

The *FuzzEqu Toolbox* includes routines for developing, tuning and testing linguistic equation systems [8]. The LE systems can be modified interactively to adapt the models to changing operating conditions. Links to *Matlab*® toolboxes facilitate comparisons with various modelling approaches, e.g. fuzzy set models, artificial neural networks, decision trees, regression models and system identification. It provides routines for building *LE* systems from large fuzzy systems including various ruleblocks implemented in FuzzyCon [9] or *Matlab*® FuzzyLogic Toolbox. Other fuzzy modelling approaches can be used as channels for combining different sources of information [4].

This paper describes new advances of the modelling and simulation of a *SuperBatch* pulp cooking. The simulation study concentrates on the heating and cooking sequence. The application is done in co-operation with *ABB* and *UPM-Kymmene* in an industrial pulp mill.

2 Steady state modelling

Linguistic equation models consist of two parts: *interactions* are handled with linear equations, and nonlinearities are taken into account by *membership definitions* [4]. The basic element is a compact equation

$$\sum_{j=1}^{m} A_{ij} X_j + B_i = 0, \qquad (1)$$

where X_j is a linguistic level for the variable j, $j = 1...m$. Linguistic values very low, low, normal, high, and very high correspond to integer numbers -2, -1, 0, 1 and 2. The direction of the interaction is represented by interaction coefficients A_{ij}. The bias term B_i was introduced for fault diagnosis systems. A *LE* model with several equations is represented as a matrix equation. Linguistic equations can be used to any direction.

The directions are usually quite clear in small systems: 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.

A membership definition is a non-linear mapping of the variable values inside its range to a certain linguistic range, usually $[-2, 2]$. The mapping is represented with two monotonous, increasing functions, which must overlap in the center at the linguistic value 0. In the present system, these functions are second order polynomials. Coefficients are extracted from data or defined on the basis of expert knowledge.

Modelling with linguistic equations has following stages:

- Membership definitions are generated by using preprocessed data.
- Linguistic relations are obtained by nonlinear scaling.
- Linguistic equations are generated from the scaled data denoted as linguistic relations.
- Selecting equations from alternatives is based either on the overall fit or on the prediction performance.
- Tuning modifies membership definitions, linguistic equations or both to improve fitting to the training data.

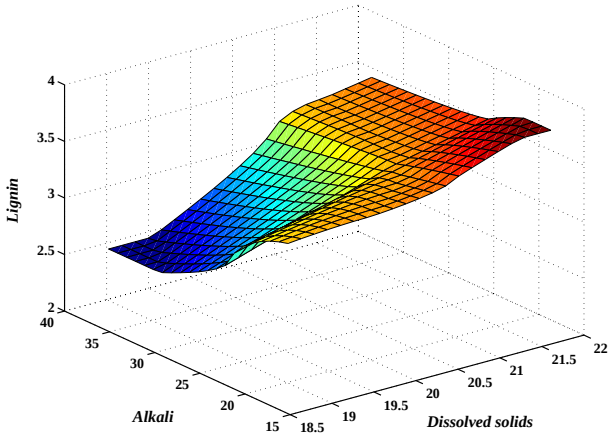

Figure 2: A single equation *LE* model for alkali, lignin, and dissolved solids.

Real-valued approach is now the main method in applications because of efficient tuning techniques. A neural network based tuning can be done for selected variables. A recently generated genetic tuning method can handle several variables at a time by varying parameters of membership definitions. For case based model development, the training data consist of several data sets. Fuzzy C-Means Clustering is used for finding these overlapping operating areas [8]. Alternatively the operating areas can be obtained by subtractive clustering, self-organizing maps (SOM) or radial basis functions. The interaction matrix is normally the same for all working areas, which is quite reasonable since the directions of interactions do not change considerably between different working points. The differences between the models are handled with membership definitions.

The modelling technique can be extended to several equations as well, e.g. by using Takagi-Sugeno (TS) type fuzzy models together with ANFIS method for development of local linear models for different operating areas. As *LE* models are non-linear, also these local models are non-linear [5].

Steady-state models were developed for interactions between alkali, lignin and dissolved solids. A single equation model shown in 2 was very accurate but test campaign specific [5]. The retuning was necessary because of drastic changes in the operating conditions, especially in properties of the cooking liquour. Two equation models based modified TS-models had similar a similar performance.

3 Dynamic modelling

Intelligent dynamic models can be constructed on the basis of state-space models, input-output models or semi-mechanistic models [4]. The most common structure for the input-output models is the NARX /Nonlinear AutoRegressive with eXogenous input) model which establishes a relation between the collection of past input-output data and the predicted output. Multiple input, multiple output (MIMO) systems can be built as a set of coupled multiple input, single output MISO models.

Delays are taken into account by moving the values of input variables correspondingly. The external dynamic models provide the dynamic behaviour. Effective delays depend on the working conditions (process case); e.g. the delays are closely related to the production rate in many industrial process. Initial estimates of the delays can be developed by correlation analysis, but similarities detected by the correlation analysis can be accidental in some cases. The delays should be assessed against process knowledge, especially if normal on-line process data is used [4]. An appropriate handling of delays extends the operating area of the model considerably.

3.1 Single model approach

In the single model approach, also variables affecting to the working point of the model are in the dynamic model equation. The development of the models proceeds as in the steady state case with some extensions. Dynamic variables include values from different time steps, or changes of the these variables are introduced as new variables. The whole dynamic model can be presented by a single equation in small systems.

In batch cooking, the dynamic models of alkali, lignin and dissolved solids are interacting. The overall model is a set of three equations, in the present these equations are used for calculation of changes for alkali, lignin and dissolved solids. For example the alkali concentration is decreasing during the cooking, and the change in absolute figure is decreasing with increasing H-factor and decreasing alkali level as shown in Figure 3. For batch cooking, each equation is represented by a model block (Figure 4).

The basic form of the *LE* model is a static mapping, and therefore dynamic *LE* models could include several inputs and outputs originating from a single variable [4]. However, rather simple input-output models, e.g. the old value of the simulated variable and the current value of the control variable as inputs and the new value of the simulated variable as an output, can be used since nonlinearities are taken into account by membership definitions.

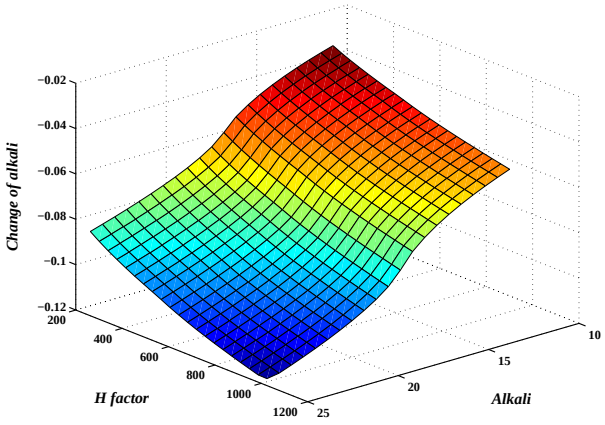

Figure 3: Model surface of the alkali model.

Comparisons with different parametric models, e.g. autoregressive moving average (*ARMAX*), autoregressive with exogeneous inputs (*ARX*), *Box-Jenkins* and Output-Error (*OE*), show that the performance improvement with additional values is negligible.

3.2 Multimodel approach

A multimodel approach based on fuzzy *LE* models has been developed for combining specialised submodels [4]. The approach is aimed for systems that cannot be sufficiently described with a single set of membership definitions because of very strong nonlinearities. Additional properties can be achieved since also equations and delays can be different in different submodels. In this approach, the working area defined by a separate working point model. For model development, the training data consist of several data sets with some overlap of the working point areas.

In batch cooking, specific submodels would be needed because of variations in the quality of chips and the properties of the incoming cooking liquor. As the differences in operating conditions affect to the speed of change, the models were adapted with a speed factor that depends on the H-factor

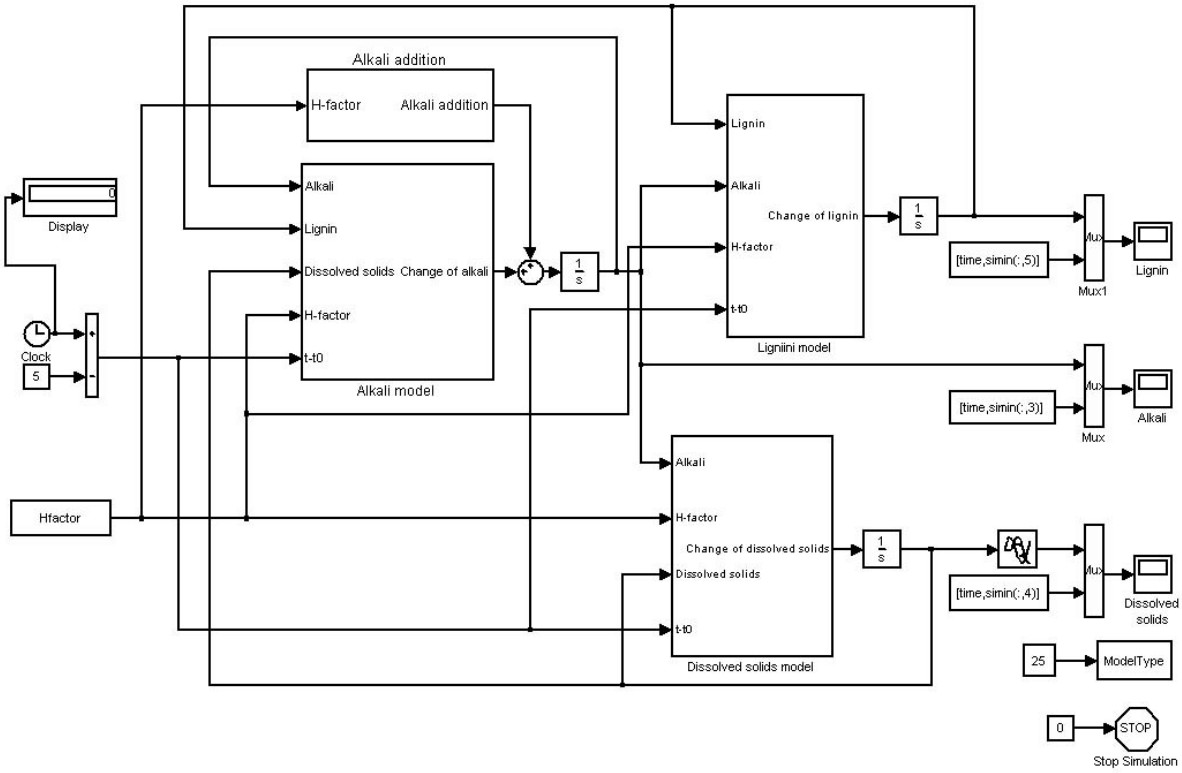

Figure 4: Simulator.

and alkali level. The clustering approach could be used if the properties of the chips and incoming cooking liquor are measured. Since these measurements are not available, the subareas must be identified by comparing the performance of the normal model.

4 Dynamic simulation

The steady-state model developed for interactions between alkali, lignin and dissolved solids was first included to a dynamic model, i.e. lingnin and dissolved solid were used in the calculation of new alkali level [5]. The dynamic model shown in Figure 4 includes two additional models: one for calculating the change of lignin, and one for calculating the change of dissolved solids. The structure of all these submodels was similar, and they were developed from the measurement data on several overlapping operating areas. The dynamic effects were handled with integration of changes.

Multimodel aspects were taken into account by adapting the model to different operating conditions with a speed factor that depends on the H-factor and alkali level. The level of the speed factor is modified to get the simulated results close to the measured values. Each submodel has its own model for the speed factor.

The simulation program was used to test the modelling performance. All the submodels were built with special blocks implemented in *Matlab-Simulink*®. Simulation results were shown in the same display together with the real data values (Figure 5). In this way, the real error between simulated and batch data was clearly seen. In the simulation, the integration methods were principally Runge-Kutta and Ode15s(stiff/NDF) with variable step size.

According to the first tests, the original model structure operated in a reasonable way [6]. Each test campaign required a new tuning for the membership definitions as each operating area was always very different from previous ones. However, the parameters generated on the basis of one day on each campaign operated fairly well for all the days of that specific campaign. The results of the test campaigns have been clustered into data sets corresponding to different operating conditions.

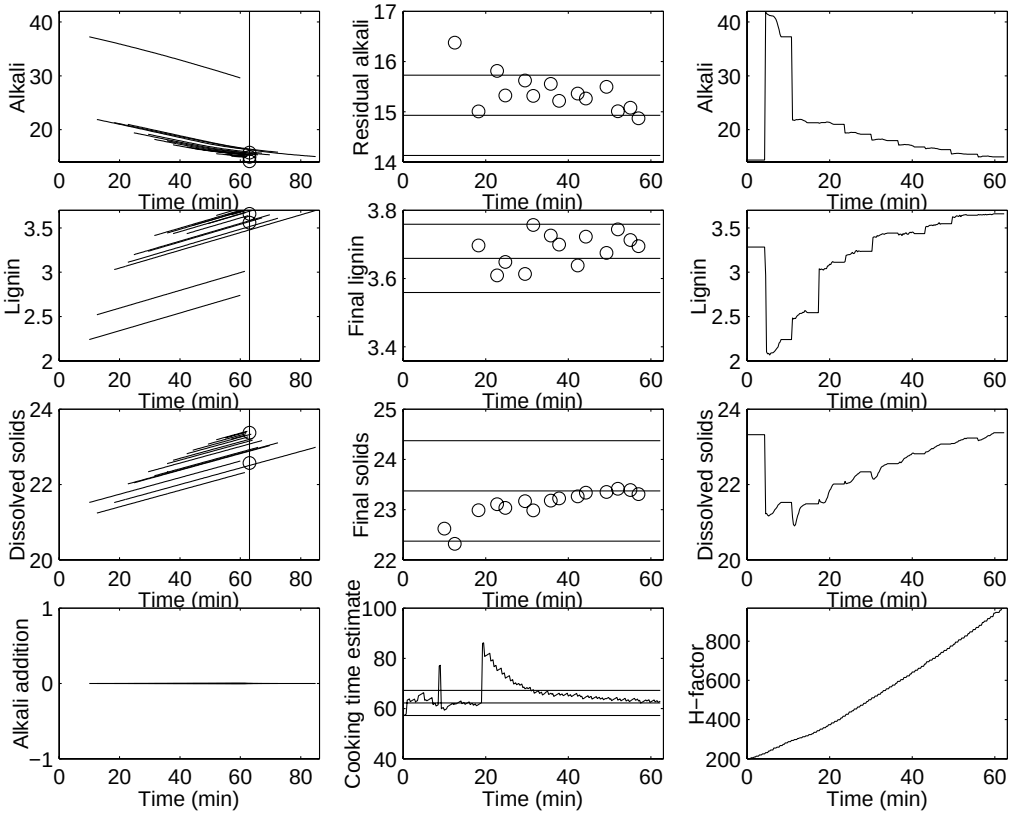

Figure 5: Test results of the simulator.

The analysis of the on-line operation changed the submodels completely, although the high level model shown in Figure 4 remained unchanged. The LE model shown in Figure 2 was not a feasible model in this case as there were all the time considerable differences in the starting levels of both the lignin and the dissolved solids. First lignin and dissolved solids were excluded in the calculation of the new alkali level, and the simulation was based direct calculation ot the alkali change. Later all the simulations have been based on calculating the changes directly.

5 On-line simulator

The dynamic simulator of the batch cooking is in on-line testing at an industrial digester house. The simulator was first tested with previous measurements. To obtain several estimates, the simulation is done to the end of cooking sequence several times during cooking process.

The measurements are not continuously updated all the time as the analyser needs a washing sequence. Therefore, the data set must be checked before starting the simulation. To emulate these features in the off-line tests, the feeding of the data was done with another *Matlab* program running simultaneously. The off-line tests are very fast: one cooking sequence takes some minutes and most of the time goes for showing the intermediate results similar to the on-line operation.

According to the test results, the speed of the change depends strongly on the operating conditions. For the first test periods, differences were too high for a reliable prediction, but later the speed factor did not need any changes for most of the cases, i.e. changes of the speed factor level mean considerable differences in the quality of chips.

The on-line simulator calculates every time the values of alkali, lignin and dissolved solids to the end of the cooking sequence on the basis of the *CLA 2000* measurements and various process measurements collected by the Damatic automation system. The simulation is started on appropriate time intervals. Figure 5 shows an example of the simulation results in the end of the cooking sequence: simulations and in the first column, pre-

dictions of the end result in the second, and actual measurements in the third. Predictions are shown after each simulations.
Good results at the early stege of the cooking sequence are promising for extending the use of the *CLA 2000* analyser and the on-line simulator to several digesters.
The on-line simulator of the cooking process is planned to be used in the control of alkali additions during the cooking sequence on the basis of the estimates of the residual alkali. The other predictions are planned to be used forecasting the quality of the pulp. In the future the aim is also to tune the controllers of the whole cooking process.

6 Conclusions

Linguistic equations have been applied in the dynamic simulation of the batch cooking process. The dynamic *LE* model handles interactions between the values of alkali, lignin and dissolved solids and can be adapted to different operating conditions with configurable parameters. The differences between the speed factors depend on other measurements related to the operating conditions. The graphical tools of *FuzzEqu* and *Simulink* have been essential in testing the quality of the dynamic *LE* models.

References

[1] K. Leiviskä. Process Control in Fiber Line. In K. Leiviskä, editor, *Process Control. Papermaking Science and Technology, Book 14*, pages 73–99. Fapet Oy, Jyväskylä, 1999.

[2] S. Murtovaara, K. Leiviskä, E. Juuso, and J. Ikäheimonen. Kappa number prediction based on cooking liquor measurements. In L. Yliniemi and E. Juuso, editors, *Proceedings of TOOLMET'99 Symposium - Tool Environments and Development Methods for Intelligent Systems, Oulu, April 15-16, 1999*, pages 210–216, Oulu, 1999. Oulun yliopistopaino.

[3] K. Leiviskä, E. Juuso, and A. Isokangas. Intelligent modelling of continuous pulp cooking. In K. Leiviskä, editor, *Industrial Applications of Soft Computing*, pages 147–158. Springer, Heidelberg, 2001.

[4] E. K. Juuso. Fuzzy control in process industry: The linguistic equation approach. In H. B. Verbruggen, H.-J. Zimmermann, and R. Babuska, editors, *Fuzzy Algorithms for Control, International Series in Intelligent Technologies*, pages 243–300. Kluwer, Boston, 1999.

[5] E. K. Juuso and E. Jäväjä. Intelligent dynamic simulation of batch cooking. In L. Yliniemi and E. Juuso, editors, *Proceedings of TOOLMET'01 Symposium - Tool Environments and Development Methods for Intelligent Systems, Oulu, April 19-20, 2001*, pages 118–129, Oulu, 2001. Oulun yliopistopaino.

[6] E. K. Juuso. Intelligent dynamic simulation of batch processes. In *Proceedings of SIMS 2001 - 42nd Conference on Simulation and Modeling, Porsgrunn, Norway, October 8-9, 2001*, pages 155–163, 2001.

[7] U. Saarela, E. Juuso, and K. Leiviskä. *Modelling of a fed-batch fermentation process. Report A 21, June 2003.* Control Engineering Laboratory, University of Oulu, Oulu, 2003.

[8] E. K. Juuso. Linguistic equations for data analysis: Fuzzequ toolbox. In L. Yliniemi and E. Juuso, editors, *Proceedings of TOOLMET 2000 Symposium - Tool Environments and Development Methods for Intelligent Systems, Oulu, April 13-14, 2000*, pages 212–226, Oulu, 2000. Oulun yliopistopaino.

[9] E. K. Juuso, J. Myllyneva, and K. Leiviskä. Fuzzy logic controller and adaptive tuning. In A. Guasch and R. M. Huber, editors, *Proceedings of the Conference on Modelling and Simulation 1994 Barcelona, Spain, June 1 -3, 1994*, pages 475–479, San Diego, USA, 1994. SCS International.

Acknowledgements

The research on batch cooking was done in co-operation with ABB and UPM Kymmene Wisaforest in KUHA project financed by The Finnish Technology Development Centre (TEKES) within the technology programme Process Integration.