8 Informatics in Solid-state Fermentation

Wilerson Sturm, Carlos R Soccol,
Dario Eduardo Amaral Dergint,
Jose A Rodríguez-León &
Deiva Canali Navarro Vieira Magalhães

8.1 INTRODUCTION

The technological progress in informatics and computation sciences has been increasing faster than most could imagine or even understand, not only because of its velocity but essentially because of its large scope of applications. With constant improvements of performance features and relative drop of prices, these equipments are more and more cost effective. One of the areas where the demand for more computing power is most notorious is in biotechnology, especially in DNA researches, where it could turn into the driving force of development because of the need to store and analyze the enormous amount of genomic and proteomic data. However, almost the same has been taken place in all other biotechnology areas. Actually the advances are not only in hardware specifications. Indeed there are new programming techniques, new and powerful languages, and even new entire programming frameworks. Together with the hardware performance these software developments allow more complex problems solution. This chapter will discuss some of the new techniques, solutions and challenges concerning the computing uses in biotechnology and show some examples of implementation like Fersol2 software, for instance.

8.2 PROGRAMMING LANGUAGES

A major problem to design simulation system is the task of choosing an adequate programming language. Normally, it is necessary to make equilibrium between the following criteria that are considered more important in this situation: rapid implementation, normally to analyze non complex problem; independence of operational system (OS), normally Windows or Linux; velocity, when the algorithms are CPU intensive; flexibility, simulating and controlling a real-time system; evolutive, developed blocks are easily re-utilizable by objects; diffused, easy to find skilled people to use a language. Here the languages analyzed are C++, C#, Java and VB (Visual Basic) which fit with the criteria presented.

Criteria\Language	C++	C#	Java	VB
Rapid implementation	–	+	+	++
Independence of OS	–	++	++	–
CPU intensive	++	++	+	–
Flexibility (simulation/real control)	++	++	+	–
Evolutive (object oriented)	+	++	++	–
Diffused	+	+	++	++

Obs.: More positive is better: – single negative, + single positive, ++ double positive.

In the context of biological simulations, VB are most utilized to make simple analysis. However, if the objective is to make evolutive platform, languages like C++, C# and Java must be considered. But, with the above criteria, the best language is C#. This language is new, but has some interesting characteristics: JAVA like, C++ like, and open standard. This hybrid nature makes C# flexible and powerful to develop virtually any application.

8.2.1 Agent and multi-agent system

Conjugated with the language, it is important to adopt a flexible and evolutionary methodology to develop the biotechnological simulating / controlling system. According to Jennings et al., (1998), the agents autonomous and the multi-agent systems represent a new manner of analyzing, projecting, and implementing complex systems by software. This approach is made up of a very powerful whole of tools, techniques and metaphors which are very interesting for the development of software. According to this approach "An agent is a computer system, situated in some environment, which is capable of flexible autonomous action in order to meet its design objectives". This work intends to check this approach potentiality for the study of biological problems.

The importance of the multiagent system lies in its capacity of complex problems resolution, that use to be difficult to identify and describe perfectly. The multiagent system is based on a certain number of agents which try to give a good response for a specific situation. To achieve that, they can exchange information and also be able to learn "how to move". The solution is a system evolution function, which is "itself" the result of a process of training. The training can be seen in two different levels: individual or by the system.

The best manner of solving a complex, large and/or unforeseeable problem is to develop a number of specific functionalities and components modular which are specialized in the resolution of a particular aspect of the problem. Here, the modular or almost modular components are the agents. Another characteristic of real problems is that they are open and distributed systems. In such a situation, it is impossible to know its components in advance and moreover, those behavior changes with time.

Sycara (1998) defined a Multi-Agent System (MAS) as a loosely coupled network of problem solvers that worked together to solve problems that were beyond the individual knowledge or capabilities of each problem solver. Nowadays, the most problems solution is based on single agent. However, the MAS has a great difference; it is not necessary to know exactly what the system needs to do before its implementation. The system can be improved with a little effort when it turns necessary or when a solution is reached.

The multi-agent system allows simulations closer to the real world, because it is possible to take into account a great diversity of variables and to articulate them to check the results. The constraints of this method are related to the system project and the results comprehension. Accordance to Dergint (1999), due to the system complex nature, it is still not easy to know or determine analytically the conclusion or the results obtained automatically. For instance, biological control and monitoring systems with MAS can use a heterogeneous strategy to find a solution, it gives characteristics as robustness and flexibility.

8.3 MODELING AND SIMULATION

Accordingly to Harrington and Tumay (2000), computer simulation modeling allows to see the results of a process without investing time and resources to test the real one. Most of simulations are able to show results with a kind of time acceleration. It, thus, becomes clear that it brings the possibility of saving resources and time. However, it demands more complete mathematical models describing the process behavior. Instead of committing resources to implement a new real process, simulation allows to test some theories previously such as changing process parameters, process scale, and initial conditions, without the real costs and specially the whole kind of risks. In addition, there is the possibility of turning the simulation model into a control solution or, at least, part of that.

8.3.1 Examples of techniques

There are many strategies for modeling and simulating industrial processes. Bioprocesses specifically have some peculiarities such as the uncertainties in the model parameters, normally described by ordinary differential equations, taking into account kinetic coefficients, for instance, and some not totally described relations as the yield and specific growth rate, depending on the product and microorganism involved, besides the processes behavior usually contain non-linearities. These characteristics lead to discuss about the importance of making use of modern mathematical modeling techniques in accordance to Edelstein-Keshet (2004). In some cases it is unfeasible to take into consideration all the variables and its behavior, or even know all of them, hence new approaches are indispensable to face these kind of problem. The following sections show several examples of useful methods.

8.3.1.1 Fuzzy logic

The concept of Fuzzy Logic, accordingly to Zadeh (1965), was presented at first time not as a control methodology, but as a way of processing data, considering partial set membership rather than crisp set membership or non-membership, as the so-called Classical Logic. As presented in Zadeh (1990), Professor Lotfi A. Zadeh explained that people do not require precise numerical information input, and yet were capable of highly adaptive control. Human beings often consider imprecise expressions like "tall", "expensive", or "not so far" (Sturm, 2005). However, the processing mode of a computer is limited to a one-or-zero, everything-or-nothing, or true-false mode of thinking. Gupta (1977) asserts that Fuzzy Logic is able to remove barriers between human being communication mode and computer processing mode. Professor Zadeh proposed the concept of linguistic or "fuzzy" variables, which invited to think as linguistic objects or words instead of numbers.

Fuzzy Logic presents a rule-based approach to describe the process behavior rather than attempting to model the system mathematically or the modeling task is empirically-based, depending on human experience and not exactly technical understanding of the whole process. This approach allows to handle poor defined problems, where there are variables not exactly known, concerning to its behavior or importance to some specific process, which frequently happens in bioprocesses. Thus, it is not necessary to consider all the variables; in most cases it could be prohibitive and the nonlinearities are often absorbed by the proper Fuzzy Logic manner of describing problems and working with them.

8.3.1.2 Neural networks

According to Norgaard et al., (2000), most of human abilities were learned by examples. That is how emerged the interest of transferring this didactic model into a computer's program. Neural Networks are a kind of approach designed to mimic the real human brain neural connections. Anthony and Bartlett (1999) mentioned that an Artificial Neural Network needs a training process, where examples were presented and were supposed to be learned. A neural network consists of a large number of simple neurons, which is often richly interconnected and each one is a part of program with normally very simple function to execute. The so-called synaptic weight associated with the connections between neurons will determine the characteristics of the network. During the training process, the network automatically adjusts the values of synaptic weights. Actually, only one neuron has no function but the pattern of synaptic weight values in the whole network that carries information. The patterns are generated by the network and, because of that, it is not possible to describe analytically the whole process used to solve the problem.

Norgaard et al., (2000) explains that the interest in using this approach is because of the ability of working with non-linear systems and the learning capacity. Sometimes researchers know the behavior of a specific Bioprocess, but are not able to describe each variable influence itself. Thus a Neural Network could be able to learn from the practical examples showed, even if the mathematical model is not understandable as an equation or something like that, but presenting the same results as the real process or manipulating the control parameters, leading to an adequate and stable work value.

8.3.1.3 Genetic algorithms (GA)

According to the theory suggested by Charles Darwin in 1859, the biological alive species evolution would be due to the natural selection resulting from the competition for food and the reproduction ("struggle for life"). Holland et al., (1986) asserts that this mechanism of creation/selection is very simple, therefore, its application as software algorithms seems to be natural. The introduction of these mechanisms on computer solutions took place in early seventies and received the name of Genetic Algorithms known by the initials GA. The way that genetic algorithms treat a problem is very simple. Firstly, GA works in prospecting on a group of objects, not on an object. The group of objects is a whole population of different objects. Each element of the group receives a fitness value. This value allows the object to know which is the best solution to a given problem and the fitness is calculated in an objective way by the means of an evaluation function. This evaluation function is the process part which is guided by a known field. However, the process of prospecting is not guided in a specific knowledge field. The individuals or objects are coded by sequences of bits so that it is possible to make an analogy with the chromosomes. As in the biological creatures, the genetic operations on these sequences of bits are not found on a specific field of knowledge. The characteristics of the individuals are coded by chromosomes neutral sequences and the genetic operations are made on these chromosomes. Thanks to this characteristic of specific science field independence, the genetic algorithms are universal and of special interest to prospect in areas whose the results possibilities are not known. With a population and their fitness a new generation is created. The process of creation analogically follows the natural process of evolution where the most powerful individuals have more chance to reproduce. Rising generation, thus, will replace, with a certain probability, the old individuals less adapted to their medium. GA turns possible to prospect in most of the target medium and in a short lapse of time. Normally the prospecting must finish when an optimum is found in target space. The probability of having found a good optimum increases accordingly to the number of individuals used in simulation.

8.4 THE FERSOL2 EXAMPLE

The new software Fersol2 is based upon the so-called Fersol. This software was developed in 1987, running only on DOS operational system, using the methods described by in Rodriguez Leon et al., (1988) to manipulate solid-state fermentation (SSF) parameters. In SSF, instead of what happens in submerged fermentation, the kinetic process behavior is very difficult to analyze. It is due to the impossible separation of biomass from the solid substrate and because of its heterogeneous characteristic (Pandey et al., 2001). The interest in studying kinetic process behavior is because of the fact that it allows many important parameters determination such as specific growth rate, process yield and productivity, heat evolved, process control criteria, strategy for the production of a particular product, and scale-up considerations (Pandey et al., 2001).

The biomass synthesis is one of the most important patterns present in a fermentation bioprocess. In submerged fermentation process, this parameter can be normally measured through direct methods such as cell counting, dry biomass or optical density determination. Nevertheless, in SSF the proper characteristic is that the biomass will be attached to solid surface, hence, it is not so easy to measure directly. Thus, some indirect methods can be used as the determination of protein content or another cell component (Raimbault 1981; Okasaki et al., 1980; Matchman et al., 1985), or through the oxygen consumption and production of CO_2 in (Nishio et al., 1979; Carrizalez et al., 1981; Koba et al., 1986). Regarding the software solution, the O_2 consumed and/or CO_2 evolved seems to be more adequate because of on-line measurement possibility and fast results (Pandey et al., 2001) and this method could be considered as a direct measurement of process kinetics, although in true sense it is still an indirect measure of biomass synthesis.

To develop the Fersol2 software, it was decided that it should run on Microsoft Windows operational system because of the advantages and resources offered by the system. It was decided to use DotNet platform and C Sharp (C#) programming language, both with new concepts of programming, because of the performance, resources, and longevity concerning the time to turn obsolete. The platform allows to develop solutions which runs under Windows and Linux operational systems, adding more versatility.

8.4.1 Fersol2 calculations

As pointed out in Pandey et al., (2001) and observing all the considerations, using some computing power it is possible to estimate the biomass (Xn) through an appropriate software solution. Making use of a mathematical set of method called Numerical Calculus, Fersol2 is able to calculate the integral and differential equations. As an example to evaluate the total oxygen consumed, it is useful the integral equation described by Pandey et al., (2001):

$$\int_{t=o}^{t=n} \frac{dO2}{dt} dt$$

From the Numerical Calculus, the numerical integration is applied by the Trapeze Rule, which leads to a discrimination of the theoretical equation, but in this specific case the values are taken from a table of oxygen quantity consumed, hence it is not quite a problem. The result is equivalent to the area under the oxygen consumption curve, determined by some periodic measures.

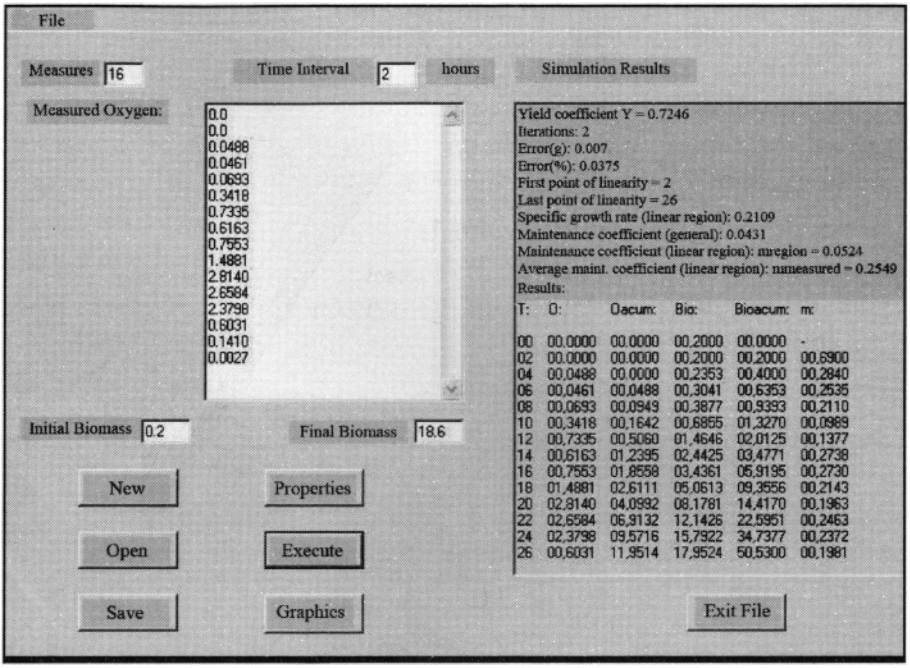
On the other hand, some problems need an iterative method to solve. To determinate the biomass (Xn), making use of Metabolic Gas Balance Method described in Pandey et al., (2001), it is necessary to make an adjust in the biomass curve using the following equation:

$$X_n = \frac{\left\langle Y_{X/O}\Delta t \left\{ \frac{1}{2}\left[\left(\frac{dO_2}{dt}\right)_{t=0} + \left(\frac{dO_2}{dt}\right)_{t=n} \right] + \sum_{i=1}^{i=n-1} \left(\frac{dO_2}{dt}\right)_{t=i} \right\} + \left(1 - \frac{a}{2}\right) X_0 - a \sum_{i=1}^{i=n-1} X_i \right\rangle}{\left(1 + \frac{a}{2}\right)}$$

Where a = m.Yx/o.Δt. Taking into account some real measures of the biomass at a determined time and the initial process conditions, a kind of trial and error method is used to adjust the intermediary values. Numerical Calculus presents many iterative methods to solve this kind of problem because the trials should not be chosen randomly, it is necessary to maximize the convergence and it allows using some stop criteria to the number of iterations. In the present time the available computers, concerning to performance features, and the software development tools allow this kind of calculation without any additional care. Instead, some years ago the programmers should keep in mind the limitations of computing power, thus the algorithms were developed to adequate the problem complexity to the time to solve it, in other words there were some problems which could be unfeasible since would take much long to solve.

8.4.2 Fersol2 results

As has been pointed out in the previous section, the Fersol2 software uses updated techniques to manipulate all the parameters. After adjusting the biomass values, it estimates many other process indicators such as, process yield based on consumed oxygen for biomass synthesis, specific growth rate at the logarithmic or exponential phase, and maintenance coefficient. It is possible to show virtually in only one window all the parameters, it turn easier to visualize the process behavior. The software interface is shown in the following figure.



It is possible to plot the curves of oxygen, biomass, and logarithmic biomass. The following figure 1 and figure 2 show the oxygen and the biomass curves.

Using the Windows or Linux operational system and additional software as text editors or spreadsheet applications software as Microsoft Office's Excel and Word or the contents of Open Office, for instance, Fersol2 allows to export or import all the values and graphics; thus, it turns easier to analyze and produce documentation.

8.5 THE DATA ACQUISITION AND CONTROL SOFTWARE EXAMPLE

Some of considerations pointed out in the previous section take into account the necessity of having on-line information about the fermentation process behavior, concerning to temperature, air flow, oxygen consuming, and CO_2

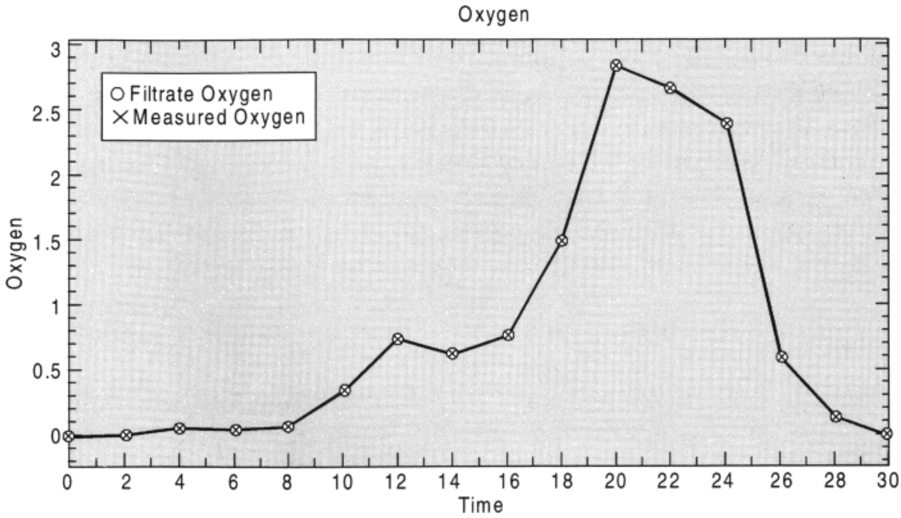


Fig. 1.

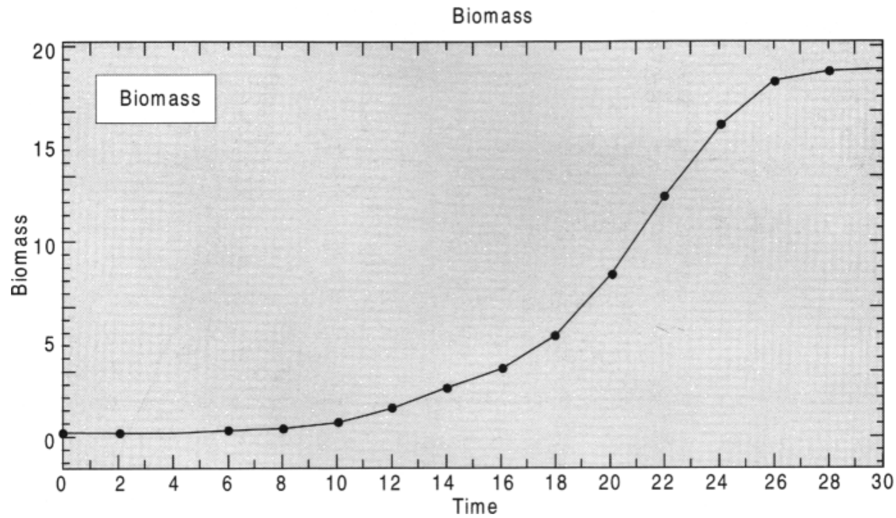


Fig. 2.

production, for instance. Sometimes to control or monitor a process it is necessary to consider real time analysis, which is all situation where there is a time window to execute some operation and/or fix some system fault, after this the error will be considered irreversible. Normally critical time applications are real time applications also. Nowadays it is possible to use the communication ports such as RS232 or, today more commonly, USB ports to make the acquisition of external sensor signals accordingly to Kelly and Bai, (2005). In addition, some protocols are getting more usual as Ethernet and Fieldbus in industrial plants and a strong trend in wireless communication as BlueTooth will certainly be present in a great number of applications. The data transfer rate has been increasing and communication protocols became more friendly specially when using the updated Windows versions and appropriate programming platform. Thus, this kind of solution turned more cost effective. The controlling devices available in industrial commerce are normally open communication protocol and the development frameworks allow making the communication drivers more friendly by the programmers.

Using adequate sensors and controlling devices, accordingly to Wang and Tan (2006), an acquisition or control software can read any parameter value and transfer it into the computer, besides the distributed control possibility allows to keep these parameters monitored even if the communication between process and computer is interrupted. In this case with eventual loss of data, of course, but keeping the advantage of maintaining the system under local control. Acquisition or control software can also display values on the screen such as tables and graphics in real time and making some calculations such as unit conversion and values statistics evaluation in accordance to Wang and Tan (2006).

8.6 CONCLUSIONS

The Solid State Fermentation process carries many peculiarities that lead to the necessity of investigating new approaches to solve several kinds of complex problems. The technological advance, in informatics and computation sciences, allows the researchers and developers to use updated techniques with even more increased performance and resources availability. These characteristics permit using some sophisticated methods with cost effective performance. An evolutive solution software could be adequate because of its flexibility and adaptation capacity. Some of the solutions discussed demonstrate that the proper indirect measures necessity in SSF is, itself, a strong argument to make use of more complete mathematical models, with the consequent need to apply more accurate analysis, which is only possible using the informatics features available nowadays.

Acknowledgment

The authors acknowledge Conselho Nacional de Desenvolvimento Cientifico e Tecnologico –CNPq for financial support and Universidade Federal do Parana – UFPR.

References

Anthony, Martin, Bartlett & Peter L, 1999, Neural Network Learning: Theoretical Foundations. 403p. p 1,5. Cambridge University Press: Cambridge.

Carrizalez V, Rodríguez H & Sardiña I, 1981, Determination of the specific growth rate of molds on semi-solid cultures. *Biotechnology and Bioengineering*, 23, 321.

Dergint Dario EA, 1999, Apprentissage Collectif et Milieux Innovateurs : Étude de Cas α Grenoble et Simulations Multi-Agents – Thesis. p 567. UTC, Compiègne.

Edelstein-Keshet L, 2004, Mathematical Models in Biology. p 586. Siam: Philadelphia.

Gupta MM, 1977, Fuzzy Automata and Decision Processes. New York: Elsevier.

Harrington H. James, & Tumay Kerim, 2000, Simulation Modeling Methods: To Reduce Risks and Increase Performance. p 1-3. p 379. McGraw-Hill: New York.

Holland John H, Keith J Nisbett, Richard E Thagard & Paul R, 1986, Induction: Processes of Inference, Learning and Discovery. Cambridge, Mass: MIT Press.

Jennings Nicholas R, Sycara Katia & Wooldridge Michael, 1998, A Roadmap of Agent Research and Development – Autonomous Agents and Multi-Agent Systems. pp. 7-38. Boston: Kluwer Academic Publishers.

Kelly Laurie & Bai Ying, 2005, The Windows Serial Port Programming Handbook. p 4, 5. CRC Press LLC: Charlotte.

Koba Y, Feroza B, Fujio Y & Ueda S, 1986, *Journal of Fermentation Technology*, 62, 2, p. 175.

Matchman SE, Jordan BR & Wood DA, 1985, Estimation of fungal biomass by three different methods. *Applied Microbiology and Biotechnology*, 21, p. 108.

Nishio N, Tai Tai & K Nagai S, 1979, Hydrolase production by Aspergillus niger in solid state cultivation. *European Jounal of Applied Microbiology and Biotechnology*, 8, p. 263.

Norgaard M, Ravn O, Poulsen NK & Hansen LK, 2000, Neural Networks for Modelling and Control of Dynamic Systems: A Practitioner's Handbook. 246 p. Springer-Verlag London Limited: Great Britain.

Okasaki N, Sugama S & Tanaka T, 1980, Mathematical model of surface culture of koji mold, Journal of Fermentation Technology, 58, 5, p. 471.

Pandey A, Soccol CR, Rodríguez-león JA & Nigam P, 2001; Solid State Fermentation in Biotechnology: Fundamentals and Applications. Asiatech Publishers, Inc. New Dehli. India.

Raimbault M, 1981, Travaux et Documents de l'ORSTOM, No. 27, París.

Rodríguez León J A, Sastre L, Echevarría J, Delgado G & Bechstedt W, 1988, A Mathematical Approach for the estimation of biomass production rate in solid-state fermentation, *Acta Biotechnologica*, 8 (4), 299-302,

Sturm Wilerson, 2005, Avaliação do Potencial de Uso da Lógica Fuzzy para a Identificação de Indicadores de Competências no Currículo Lattes. Dissertation. p 104. April-25-2005. Centro Federal de Educação Tecnológica do Paraná. Curitiba.

Sycara Katia P, 1998, Multiagent Systems, AI Magazine, 79-92.

Wang Lingfeng, Tan & Kay Chen, 2006, Industrial Automation Software Design, p 158,159 John Wiley and Sons: New Jersey.

Zadeh LA, 1965, Fuzzy Sets. *Information and Control*, 8, 338-353.

Zadeh LA, 1990, The birth and evolution of fuzzy logic. In: TURKSEN, I.B. Proceedings of NAFIP'90.