Process control and artificial intelligence software for aquaculture

Phillip G. Lee a,b,*

a Marine Biomedical Institute, University of Texas Medical Branch, Galveston, TX 77555-1163, USA
b Aquatic Feeds and Filters, Inc., PO Box 3131, Galveston, TX 77552, USA

Received 15 August 1999; accepted 17 February 2000

Abstract

Modern research and commercial aquaculture operations have begun to adopt new technologies, including computer control systems. Aquaculturists realize that by controlling the environmental conditions and system inputs (e.g. water, oxygen, temperature, feed rate and stocking density), physiological rates of cultured species and final process outputs (e.g. ammonia, pH and growth) can be regulated. These are exactly the kinds of practical measurements that will allow commercial aquaculture facilities to optimize their efficiency by reducing labor and utility costs. Anticipated benefits for aquaculture process control and artificial intelligence systems are: (1) increased process efficiency; (2) reduced energy and water losses; (3) reduced labor costs; (4) reduced stress and disease; (5) improved accounting; and (6) improved understanding of the process. This review explores the technologies and implementation of the technologies necessary for the development of computer intelligent management systems for enhanced commercial aquaculture production. Today’s artificial intelligence (AI) systems (i.e. expert systems and neural networks) offer the aquaculturist a proven methodology for implementing management systems that are both intuitive and inferential. There have been many successful commercial applications of AI (e.g. expert systems in cameras and automobiles). The major factors to consider in the design and purchase of process control and artificial intelligence software are functionality/intuitiveness, compatibility, flexibility, upgrade path, hardware requirements and cost. Of these, intuitiveness and compatibility are the most important. The software must be intuitive to the user or they will not use the system. Regarding compatibility, the manufacturer should be congruent with open architecture designs so that the chosen software is interchangeable with other software products. © 2000 Elsevier Science B.V. All rights reserved.

* Tel.: + 1-409-7722133; fax: + 1-409-7726993.
E-mail address: pglee@utmb.edu (P.G. Lee)
Keywords: Automation; Process control; Artificial intelligence; Expert system; Fuzzy logic; Neural network

1. Introduction

The field of aquaculture has been considered by many of its practitioners to be as much of an art as a science; success of aquaculture operations has been closely associated with the intuition of the farm manager rather than with understanding of the physiology, ecology and behavior of the cultured species. Hence, farm managers have been hesitant to trust their crops to automated management systems. However, recent research and commercial operations have begun to adopt new technologies and aquaculture as a science is evolving (Lee, 1995; Malone and DeLosReyes, 1997). Real-time trending of system parameters provides the manager with unprecedented insights into the physical and biological conditions of the aquaculture facility. These insights would be impossible with manually monitored aquaculture systems because of the labor required to collect and enter data and then to prepare graphs and reports. Recirculating aquaculture systems have the most obvious needs for this technology but pond and offshore aquaculture systems can benefit as well.

2. Advantages and benefits of process control systems

The purpose for applying process control technology to aquaculture in developed countries encompasses many socioeconomic factors, including variable climate, high labor costs, increased competition for dwindling water and land resources and an unsympathetic regulatory bureaucracy. These factors are pushing the US and other developed nations toward the use of intensive, recirculating, water filtration systems and off-shore pens and cages (Fridley, 1993; McCoy, 1993; Lee, 1995; Hayden, 1997; Helsley, 1997). High efficiency, automated control systems should (1) reduce simultaneously the need for high quality make-up water and the volume of pollutant-laden effluent for land-based recirculating systems, and (2) reduce labor costs for on-site supervision and normal feed wastage associated with off-shore aquaculture (Lee, 1995). Anticipated benefits for aquaculture process control systems are: (1) increased process efficiency; (2) reduced energy and water losses; (3) reduced labor costs; (4) reduced stress and disease; (5) improved accounting; and (6) improved understanding of the process.

The use of computer monitoring and automation in aquaculture has grown during the last 15 years. Applications include algae and food production (Rusch and Malone, 1991, 1993), feed management (Hoy, 1985), filtration systems (Whitson et al., 1993; Lee et al., 1995, 1999; Turk et al., 1997), vision systems (Whitsell and Lee, 1994; Whitsell et al., 1997), environmental monitoring and control (Hansen, 1987; Ebeling, 1991, 1993; Munasinghe et al., 1993) and integrated system
management (Lee, 1991, 1993; Lee et al., 1995, 1999; Turk et al., 1997). This migration toward intensification and automation parallels the development of other forms of agriculture, such as, automated broiler coops (Campbell, 1988; Allison et al., 1991), nursery greenhouses (Bakker et al., 1988; Hooper, 1988; Jones et al., 1990), dairy barns and feedlots (Leonard and McQuitty, 1982; Mottram and Street 1991) and crop irrigation systems (Rao et al., 1992). These intensive agricultural production systems share many characteristics with intensive aquaculture systems and all are commodity markets (i.e. high production and low profit margins).

Process control technology is big business; control system integration revenues were expected to reach US$10 billion in 1998 and the industry has been growing at a rate of > 15%/year over the last 5 years (Kuhfeld, 1994).

The application of process control technology to aquaculture systems will allow the aquaculture industry to: (1) site production closer to markets; (2) improve environmental control; (3) reduce catastrophic losses; (4) avoid problems with environmental regulations; (5) reduce management and labor costs; and (6) improve product quality and consistency (Lee, 1995). The overall design and implementation (i.e. hardware and software) of aquaculture process control systems has been reviewed (Lee, 1991, 1995). These reviews included both custom-designed (i.e. user-designed) and commercially available systems. Custom-designed systems are usually designed and constructed by one individual who is technically competent with computer hardware and software; these custom systems often require thousands of lines of computer code to be written. This is unnecessary today because there are many cost-effective process control software programs and compatible hardware available on the market. Therefore, this review focuses specifically on the characteristics and capabilities of commercial process control and artificial intelligence software.

3. Process control software characteristics

3.1. Operating systems

The choices of computer platform and operating systems are the first choices to be made in selecting a process control system (Wolske, 1989; Chandler, 1994; Labs, 1994). The most appropriate criteria to use for selection are: (1) functionality or suitability to task; (2) compatibility and interconnectivity; (3) architectural expandability; and (4) price performance. There are essentially four operating systems used by process control integrators today, AT&T’s UNIX®, IBM’s OS/2®, DEC’s VMS® and Microsoft’s Windows NT™. Many companies make process control systems for all of these operating systems although Windows NT™ systems are by far the most common due to market acceptance. All of these operating systems are true 32-bit multitasking operating systems that run on PC compatible hardware systems (Chandler 1994). There are MacIntosh™ compatible process control systems available today but they are far less common because the initial capital costs are usually much higher. Recent trends indicate that this cost differential will be less in the future.
3.2. System architecture

There are four basic designs for automated control systems used in aquaculture (Lee, 1995): (1) closed loop controller or data logger systems (Losordo et al., 1988; Madenjian et al., 1988; Green and Teichert-Coddington, 1991; Ebeling, 1991; Lyon et al., 1993); (2) programmable logic controller (PLC) systems (Padala and Zilber, 1991; Ebeling, 1993; Widmyer and Widmyer, 1993); (3) microcomputer-based supervisory control and data acquisition (SCADA) systems (Hoy, 1985; Ruohonen, 1987; Lee, 1991, 1993; Rusch and Malone, 1991, 1993; Whitson et al., 1993); and (4) distributed control systems (DCS) (Hansen, 1987; Lee et al., 1995). Each design has its advantages and limitations.

A SCADA system costs significantly less than a DCS but it sacrifices redundancy needed for some critical functions (e.g. oxygen control) (Yingst, 1988). Hence, a SCADA system may be adequate for small aquaculture facilities or it may be a logical starting point for the installation of a control system in a large facility. However, larger facilities will require a DCS due to their more complex control problems, requiring that multiple intelligent devices (i.e. multiplexers, PLCs or PCs) be integrated for a higher level of control (Grenier, 1994). While DCSs were expensive in the past, the use of personal computers has reduced these costs, improved the compatibility and simplified the operation of the DCS dramatically (Lee, 1995). In recent years, distinctions between PLC, SCADA and DCS systems are becoming less clear with the establishment of open architectures and improved compatibility (Spennato and Noblett, 1992). All of these designs require compatible software to integrate their function; choice of industry standard, open architecture components will speed implementation and reduce costs.

3.3. Modules

A typical process control software application is composed of various modules (sub-routines) that perform the needed tasks (Intellution, 1994). These modules include communications, database management, graphical interface and control algorithm functions. The first four modules discussed below are the basic building blocks for a process control system (i.e. I/O drivers; database builder, scan, alarm and control; and man-machine interface) while the others that are discussed below are enhancements. The figures (Figs. 1–4) used to illustrate the following software modules are actual screens used in the set-up and operation of a computer automated denitrifying biofilter (bioreactor) (Whitson et al., 1993; Lee et al., 1995; Turk et al., 1997; Lea et al., 1998).

3.3.1. I/O drivers

The first thing that process control software must do is acquire information from the process (i.e. analog and digital inputs and outputs; I/O). The software should provide a wide variety of I/O drivers that support industry standard I/O devices each of which will have its own specific protocols and reference manuals. These industry standard devices include intelligent sensors, meters, data multiplexers or
PLCs. For each specific I/O device there is a specific I/O driver that should provide capabilities for automatic communication error detection, reporting and recovery, and support for redundant communications. The most common of these devices are data multiplexers which are usually non-intelligent signal conditioners (e.g. convert digital and analog voltages into 8-bit digital data) and PLCs which are intelligent and capable of control functions. Finally, the I/O driver is used to build and update the Driver Image Table, DIT (i.e. poll table) that can be thought of as a collection of transient mailboxes (registers) containing a single data point or range of data points.

3.3.2. Process database builder

The heart of the process control software is the process database, a representation of the process created using process control logic. The process database consists of blocks (i.e. coded set of process instructions that perform a specific task) and chains (i.e. series of connected blocks that create a monitoring or control loop) (Fig. 1). There are two general types of blocks; primary blocks that read data from or write data to the DIT and secondary blocks that manipulate data passed to them by a primary block. Blocks have also referred to as tags in certain software. Most process paradigms require several chains to be created. Typical input fields for an

![Fig. 1. This database block is an Analog Input Block for dissolved oxygen concentration in the sparge column of the nutrition system bioreactor (NUTAN). It has fields for tag name, description, device and I/O addresses, engineering units, alarm set points and security areas. For more details, see the text. In the background are additional tag names for other database blocks.](image-url)
input block include the tag name, description, scan time, PO address, signal conditioning, engineering units and calibration range, alarm set points, security, and links to the next block in the chain.

Fig. 1 shows an example of a primary block that is the initial block in a chain used to control the dissolved oxygen concentration in an influent sparge column for a denitrifying bioreactor (see Fig. 2). In this example, the oxygen sensor is present at channel 10 of the NUTAN (i.e. nutrition tank system) multiplexer; the dissolved oxygen is monitored ((DENIT-DO) every 5 s. A value above the high set point value (2.00 ppm) causes a relay to open a valve to a nitrogen source (the next block; DENIT-NSPG), causing nitrogen to strip oxygen from the influent stream and keeping dissolved oxygen in the anaerobic filter below 2.0 ppm.

3.3.3. Man-machine interface (MMI)

If the database builder is the heart of process control software, then the MMI functions as the sensory system (e.g. eyes, ears and touch) because it provides the window into the process (Cleaveland, 1993; Labs, 1993). It requires output (e.g. video display and message boards) and input (e.g. mouse, track ball, keyboard, touch screen and voice activation) hardware. The MMI can be subdivided into two
subsystems. The first is the graphics application composed of a draw or paint program (graphics formatting for display screens) and a view program (real-time display of system function). Modern graphics applications are based on graphic user interfaces (GUI) that allow a user to personalize the display using a variety of tools including object-oriented programming technologies. The second sub-system is a Links program that accesses information in the database and passes this information between various displays and databases. These links can write information into databases and control display animation.

The MMI is the component of a process control system that most process managers use routinely because it allows the manager to acquire needed trending information and to interact with the process (Fig. 2). This display represents a real-time view of process function for a denitrifying bioreactor. The oxygen sparge column (on the left of screen) has a dissolved oxygen (DO) reading above 2.0 ppm (the set point) so that nitrogen stripping bubbles are animated in the column. The pump rate is set for 100 l/h and the actual flow is 101.1 l/h maintained by a PID (proportional integral derivative) controller function. The bacterial column (center of screen) is set for a water depth equal to valve three (v3). The effluent has a pH of 8.06 and oxidation–reduction potential (ORP) of $-365.5$ mV.

Fig. 3. Historical trend chart for an automated denitrifying bioreactor (Lee et al., 1995). The chart displays graphs of ORP, pH, dissolved oxygen, pump flow rate and status of nitrogen sparge valve. For more details, see the text.
Remember that you are using a personal computer, so have the software application personalized to meet your specific requirements. Some people prefer complex graphic schematics of their aquaculture system, some prefer buttons and dials, others prefer animated displays, while others just want spreadsheets of data available. The process control software should make all of these options available. Just as each manager has a preference for the display of data, they also have preferences for the use of pointing devices and/or keyboarding; the system should allow for many different input options. Future improvements in voice activation and virtual reality will improve our abilities as managers to interact with a process.

3.3.4. Scan, alarm and control program

The scan, alarm and control program (SAC) can be thought of as the nervous system which integrates the functions of the input and output interfaces (i.e. I/O drivers and man-machine interface) and databases (e.g. DIT and control logic). The SAC is responsible for executing the logic contained in the database chains described above. The SAC reads data from the DIT, translates data into required format, checks data against set points and alarms, executes control logic and writes requested data to the DIT. However, the SAC is virtually invisible to the user.
3.3.5. Historical trending

Historical trending is used to analyze process trends, archive process data, monitor efficiency of process, and analyze post-process data. The historical assign, collect and display programs allow the manager to sample real-time data from the DIT at user selected rates and then to store to disk and/or display as process displays. These three programs collectively might be thought of as the control system's memory, short-term and long-term. The historical assign program defines your collection strategy, allowing you to identify data that you want to collect (i.e. place in a database). The historical collect program collects data that is stored in disk files; the historical display program is used to display data in user-defined trend charts (Fig. 3). This display contains trend curves for ORP, DO, water flow, pump output rate, methanol input rate and sparge valve status plotted versus time in hours. This information is both displayed and stored to disk such that an operator can access data for the current day or any previous day.

3.3.6. Recipe builder

This module allows the manager to design, implement and run recipes for a process (i.e. batch processes that have standard but repetitive steps which can be programmed). It also provides a flexible method for keeping an audit trail; this is particularly useful if the process requires the operator to change many process database values frequently. Examples of aquaculture applications would be batch processes such as algae or micro-invertebrate culture. The manager would first create a master recipe using the database builder and tag names. This master recipe would embody the expertise of the algae culturist. For each particular batch of required algae, it might need to be modified somewhat based on batch size and species so that a control recipe would be formulated from the original master recipe. The same master recipe would then be used over and over again to formulate control recipes without human error or juxtaposition of steps.

3.3.7. Tag group editor

This module simplifies recipe and display management by providing a means for accessing similar database information at different times using a single picture or symbol. For example, a large aquaculture facility is generally composed of many similar tank systems in which only some variables may vary (e.g. feeding rate and temperature). It would be unnecessarily repetitious to use the database builder to configure every individual tank system. The tag group editor allows you to reduce development time by defining one display and/or one control logic chain for all the similar groups of tag names (i.e. tanks).

3.3.8. DDE server

One of the key attributes of the Windows® operating system is dynamic data exchange (DDE) in which real-time and historical data can be exchanged automatically between two DDE compliant programs, using shared memory. This exchange of data can occur on a single computer or across a network to a remote node. The ability to act as a server (application that provides data) or client (application that
receives data) to other application software opens up many possibilities for control paradigms and report generation. The manager can take advantage of sub-routines available in other software that are not available in the purchased process control software. This is the most common method used to interface report generation, phone dialer and artificial intelligence software with the process control software. In order for DDE to work, your computer’s software must have both a NetDDE® server and client mode (these are provided in Microsoft’s Windows for Workgroups™ and higher operating systems or may come as part of the process control software).

3.3.9. Relational databases interface

This interface module lets the manager collect and write real-time data from the process database to an external relational database (e.g. Access®, dBase® and Oracle®). These relational databases must conform to industry standards for open database connectivity (ODBC) and sort, query and logic (SQL). This interface also allows you to read data stored in relational databases and write it back to a process database. No modern process control system can be considered complete without this capability since it addresses both open architecture and compatibility. This database interface is essential for future compatibility with evolving process control software products (e.g. statistical process control and artificial intelligence systems).

3.3.10. Report generator

The report generator is used to collect essential or critical data into a report for use by the manager or other users to make decisions on the process efficiency. Report generation can be internal to the process control software or require a NetDDE® link so that any DDE compatible application (e.g. Excel® or Lotus®) can be used to prepare reports. Either method can be automated such that reports are printed or displayed on a video monitor with a fixed schedule. The report generator should be able to access real-time process data, historical process databases and alarm and message files. Examples of useful reports would be weekly production statistics, feed efficiency projections, weekly or daily water quality records and materials management.

3.3.11. Security

The process control software should allow the manager to assign various levels of access to different modules, monitor displays, critical program functions and databases comprising the complete process control software system. This is usually done through a password that enables a user to access certain security areas. A security area is a group of database blocks or chains with the same security level. A user who logs on in a particular security area can access view and historical trending displays or change database blocks in that area but not in other security areas. The software must then allow the manager to create user or group accounts, assign user names and passwords, assign security area names and lastly assign users rights to each security level (Fig. 4). User name pglee (the author) is assigned a password, a group membership (customer defined titles), security areas, and access
to application features. In this case, pglee has access to all security areas and applications. In practice, a technical employee might be able to access all monitor displays of the facility and acknowledge alarms for their area of responsibility but they could not make any changes to database blocks. A financial officer might only be able to access security areas dealing with materials flow and utilities while the project manager would have access to all areas, including database blocks.

3.3.12. Alarm and messaging

Process control software should have capabilities for generating, displaying and storing a variety of alarms and messages. You should be able to route these alarms or messages to any computer linked to the system (i.e. node), printers, disk-based files, alarm summary displays, message boards and phone dialers. The alarms or messages usually require an acknowledgment from the operator before they can be turned off.

There are several general types of alarms and messages included in process control software. The most common is a database block alarm in which a database block generates an alarm when input values fall outside the upper or lower set point, a change in state occurs or when communications fail. These alarms are displayed in an alarm manager display and can be routed to audible bells, message boards and/or phone dialers; they usually require acknowledgment from the user. The second is a block message that usually goes to a printer or alarm history file to document an event at that block. The third is a operator message that makes a historical record of important operator actions in the alarm history file. The fourth is a system message that generates messages associated with errors in the software. The fifth is an application message that documents activity in another software module (e.g. recipe builder and historical collect). Both system and application messages are stored in the alarm history file so that records of the problems can be reviewed by your software vendor. All of these alarms or messages are to be used to troubleshoot problems with the actual process or problems with the control system components.

3.3.13. Remote dialer

The remote software should make real-time, historical and alarm information available across your network protocol to other process control nodes in remote locations as well as modern communications with remote computers. To be most useful a remote node should be able to (1) receive real-time alarms and messages from the process node, (2) view the real-time monitor displays on the process node, (3) make changes to the database blocks, (4) retrieve and display historical data, and (5) copy files. The remote utility should have adequate security in place so that only those users with privilege for a specific security area can access the process node from a remote node. The remote system must include a dial manager program in order to communicate over regular phone lines; the preferred emulation uses leased lines for dedicated communication.
3.3.14. Statistical control

The theory of statistical process control (SPC) is based on the assumption that a process will remain stable (i.e. within statistical limits) unless an unexpected event occurs. The role of the SPC software is to resolve an event from background process noise as soon as possible, correct the process back to its stable range and identify the cause for the instability (Dybeck, 1989). There are two types of events that cause a departure from the expected, random events and assignable events. A SPC module enables the user to use statistical analyses to trend the process (i.e. control chart), setting the upper and lower control limits (usually \( \pm 3 \) SD) of the process based on any combination of input and output blocks. When the output falls within the control limits, the variability is probably due to random events. When the output falls outside of the control limits, the variability can probably be assigned to specific events. In an industrial setting it is at this point that a production line is shutdown, the problem is identified and the control logic is corrected. This is more difficult to do in a continuous process like a fish production tank since you can not simply shutdown the process. As such, additional limits may be set to trigger alarms so that the manager can begin monitoring the control chart before the process goes out-of-limits. These modules provide not only a variety of statistical analysis presentation graphics but can include control features (Wolske, 1989).

3.3.15. Simulation development

Process simulation can be used to replicate or model an existing or proposed process. It is most useful for evaluating the function of a control paradigm before implementation but can be used to fine tune or to train personnel in the operation of an existing process control system (Dowling and Sullivan, 1993). The simulation software should (1) operate on the same hardware and in the same operating system as the control system software, (2) allow development to be interactive, (3) have predefined algorithms as well as user-defined algorithms for process functions, and (4) allow implementation with minimal changes to process control system software. Interactive modeling in real-time is the key advantage of such systems (Lee, 1991); estimates of the time saved in implementing a new process control paradigm suggest that development time can be cut dramatically, using simulation software (Nisenfeld, 1989).

3.4. Basic set-up and operation

The first task is to use your specific I/O driver configuration software to specify the input data that you need to acquire for the DIT and its location. The second step is the creation of the process database, including reads and writes to the DIT, calculations, process control logic and outputs to MMI devices. The database builder is used to configure how each I/O data point and database block will be processed. The third step requires the use of the graphics application to construct the MMI, archive and access data and generate reports. Once the process database is constructed, you use a draw program to create displays (i.e. easy-to-understand pictures) of the process data. You can then use any number of other modules (e.g.
tag group editor, recipe builder, macro editor, historical collect and assign, DDE server and report generator) to customize the software to meet the manager’s requirements. Finally, the manager will define the security areas in the system and assign security levels to all employees who will be using the system.

4. Artificial intelligence (AI) software characteristics

The use of artificial intelligence (AI) has become more common in industrial and manufacturing process control systems in recent years (Bechtold, 1993, 1994; Gilles, 1995; Hirota, 1995; Takahashi, 1995). In the past, AI software was very expensive and had greatly varying functionality. The advantages of AI systems include: (1) the rapid transfer of expert knowledge throughout an industry, especially those young industries that do not have enough available experts; (2) a reduction in labor costs due to automation of all primary functions; (3) improved process stability and efficiency; and (4) improved understanding of the process through the development and testing of the rules. Their usefulness in aquaculture has been advocated due to all of these reasons (Palmer, 1989; Lee, 1993, 1995). Modern AI systems can be divided into three main classes, knowledge-based expert systems, fuzzy logic-based expert systems and neural networks (Rock and Guerin, 1992; Bechtold, 1993; Studt, 1994); each class has its best application.

4.1. Knowledge-based expert system

Knowledge-based expert systems are computer programs that mimic the actions of acknowledged process experts (Bechtold, 1993, 1994). They require defined rules

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Example rules from a knowledge-based expert system (A) and a fuzzy-logic based expert system (B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A) Example rules for a knowledge-based expert system used to monitor water flow in a recirculating system (Padala and Zilber, 1991)</td>
<td></td>
</tr>
<tr>
<td>Rule 1</td>
<td>IF NO FILTER PUMP FLOW IS PRESENT AND FILTER PUMP IS ACTIVATED AND FILTER PUMP PRESSURE IS PRESENT THEN FILTER PUMP FLOW SWITCH FAILURE</td>
</tr>
<tr>
<td>Rule 2</td>
<td>IF FILTER PUMP FLOW SWITCH FAILURE THEN ACTIVATE ALARM</td>
</tr>
<tr>
<td>(B) Example rules for a fuzzy logic-based expert system used to control the residence time in an automated denitrifying bioreactora</td>
<td></td>
</tr>
<tr>
<td>Rule 1</td>
<td>IF (ORP IS Nom) and (Δ ORP IS Z) THEN Δ PumpRate IS Z</td>
</tr>
<tr>
<td>Rule 2</td>
<td>IF (ORP IS NL) AND (Δ ORP IS P) THEN Δ PumpRate IS Z</td>
</tr>
<tr>
<td>Rule 3</td>
<td>IF (ORP IS NL) AND (Residence Time IS H) AND (Δ ORP IS NOT P) THEN Δ PumpRate IS P</td>
</tr>
<tr>
<td>Rule 4</td>
<td>IF (ORP IS H) AND (Residence Time IS L) AND (Δ ORP IS NOT N) THEN Δ PumpRate IS N</td>
</tr>
</tbody>
</table>

a The abbreviations Nom, H, L, Z, P, N and NL represent the fuzzy sets nominal, high, low, zero, positive, negative and negative low (Lea et al., 1998).
(IF and THEN statements) or graphical knowledge (flow charts or logic trees) to be formulated by process experts (Table 1). This necessitates the rather tedious task of recording a process expert’s knowledge into the form of rules and then validating the expert system against the expert’s decisions. Generally, experienced process experts find this process antagonistic, especially when they contradict the outcomes of their earlier rules. The process requires a patient expert and an even more patient computer programmer to refine or change rules. The most significant consequence of a knowledge-based expert system is that it gives a process expert the ability to quickly distribute their intelligence throughout an organization or entire industry.

Knowledge-based expert systems do have the advantage of having an explanation facility where the user can find out why a decision was made by referencing the rules used. This makes them useful for training new managers or educating novices. It is the most tedious of the AI classes to use because of the repeated need to validate the rule-base. Table 1A provides an example of two rules that when combined signal a failure in a flow switch. Rule requires three conditions to be met (all or nothing conditions) and Rule 2 activates the alarm. One can easily see the logic and use these rules to understand the control decisions made by the rule base.

4.2. Fuzzy logic-based expert system

Fuzzy logic systems do not require defined rules or knowledge but use fuzzy rule-bases that emulate human intuition rather the absolute knowledge of experienced process experts (Czogala and Rawlik, 1989; Karr, 1993). They operate similar to a knowledge-based expert system because they do require written rules (IF and THEN) and hence an expert. However, these rules are fuzzy rules (Table 1B) and not simply yes or no rules; each rule has a membership function. Fuzzy logic-based control systems seem the most appropriate system for aquaculture because fuzzy systems process information involving uncertainty, ambiguity and contradiction better.

A fuzzy set differs from classical set theory in which set membership is discrete (either ‘member’ = 0 or ‘nonmember’ = 1) in that fuzzy set membership is identified in a continuum between 0 and 1. In other words, an object with a strong relationship will have a numerical value \( \geq 1 \) while one with a weak relationship will have a value \( \leq 0 \). The simplest example is human age. A 10-year-old is young but an 8-day-old baby is younger. In other words, the baby has a stronger membership (\( \approx 1 \)) in the class young than the 10-year-old (\( \approx 0.9 \)) whereas a 90-year-old person has a very, very low membership in young (\( \approx 0.1 \) but not 0). As a result in a fuzzy logic system, the process inputs are ascribed memberships in appropriate rules (usually many rules) and these memberships are summed and then actions (process outputs) are taken (defuzzification). Based on the membership values, these actions can be equally fuzzy.

For example, if the water level in a tank is a little low then a little water is added. This kind of system usually requires repetitive actions before a given desired state is reached (i.e. targeted water level). Fuzzy systems are therefore likely to outper-
form classical knowledge based systems because of their ability to deal with continuous changes and to generate small or large control actions relative to the degree of exhibited deviation from the desired state. Table 1B lists several fuzzy rules used to control residence time in a denitrifying bioreactor, using the pump rate of the main water pump. Rules 1 and 2 define conditions where the pump rate would not be changed whereas Rules 3 and 4 define conditions where the pump rate would be increased or decreased, respectively.

### 4.3. Neural network

Neural nets behave completely different from the previously described AI systems in that they use multiple three-dimensional neurons (i.e. processing elements) to learn to control a process but require no rules (Chester, 1992). This means that no expert is required but a large and extremely accurate database must be available. This training database can be in the form of real-time process control data or historical databases. These are generally lacking for aquaculture especially for a
new site or newly cultured species; the fact that most aquaculture projects are site specific also means that neural network systems will be trained with relatively incomplete data. However, neural nets do provide a means for the automation of poorly understood processes for which little knowledge exists — a condition that occurs all too often in aquaculture operations.

Several types of modern neural networks are shown in Fig. 5A and Fig. 5B. The Process Control Neural Network (PCNN) was designed specifically for use by the process control industry and represents a net based on probabilities similar to statistical process control techniques discussed above. Hence, you can train a neural net to identify the process conditions that may lead to each out-of-control situation (Plummer, 1993). The unique features of the PCNN are the PNN Pattern Layer and the Grossberg Layer (Fig. 5A). The PNN Pattern Layer contains a vector of values, corresponding to an input variable typical of the category it represents. The proximity of each input vector to each pattern is then calculated and the output is analyzed in the subsequent Summation Layer. As a result, some categories will drop out of the analysis and clusters will be formed. Finally, the Grossberg Layer functions as a look-up table since each node in this layer represents a potential cause for every known out-of-control situation (e.g. temperature instability and water quality) (Plummer, 1993). This layer must be trained using supervised learning, requiring each potential cause for instability to be accurately identified in the training database.

The second type of neural network shown in Fig. 5B is a self-organizing map (SOM). The SOM has a multi-dimensional network structure that can, from a random starting point, find the natural relationships among patterns with no external training (Hiotis, 1993). The SOM identifies trends and adapts according to the function of the network. It improves on typical ‘backpropagation’ techniques in that the SOM Layer is fully connected to both the input and output layers instead of containing multiple independent hidden layers as in backpropagation nets. The objective of the SOM is to compress information input from a multi-dimensional array into a two-dimensional space, forming reduced representations of the most relevant facts, without loss of knowledge about their relationships (Hiotis, 1993). The SOM also has the advantage that it offers a graphical representation of the results instead of the usual ‘blackbox’ results of neural networks (Hiotis, 1993).

4.4. Hybrid systems

As with most modern computer technologies, the distinctions between these three AI systems are becoming blurred. In particular, new hybrid neural fuzzy systems are appearing. One such hybrid is an adaptive network-based fuzzy inference system (ANFIS) (Jang, 1993; Jang and Gulley, 1995). This hybrid is comprised of multiple independent decision making units (DMUs) that require an expert to conduct training sessions for neural network development, speeding the learning process significantly. The output of the DIVW's is then fed into a fuzzy inference system (FIS) composed of a matrix of fuzzy rules that when incorporated into a
programming object allows for rapid decision-making. This type of system improves on neural networks alone in that the DMUs are not simple ‘blackboxes’ as are the neurons in a neural network and the FIS allows the user to understand the internal state of the system. Finally, the rules that make up the FIS can be modified by a process expert, improving accuracy.

4.5. Aquaculture AI systems

Expert systems (Padala and Zilber, 1991), fuzzy logic controllers (Turk et al., 1997; Lea et al., 1998) and a hybrid neural fuzzy system (Whitsell et al., 1997) have been used to automate aquaculture environmental control, water filtration and computer vision systems, respectively. Padala and Zilber (1991) designed a recirculating intensive aquaculture expert (RIAX) to manage an intensive *Tilapia* (*Tilapia mozambique*) culture system. The RIAX was a rule-based expert system designed to optimize efficiency and production by monitoring and controlling feeding, temperature, water quality, flow, oxygen and water level (Table 1A). The system was composed of sensors, meters, and actuators (e.g. valves and pumps) connected to a PLC with video monitor and uninterruptible power supply. The expert system was composed of four knowledge bases, the startup knowledge base, the run time knowledge base, the operator-assisted maintenance knowledge base and the equipment diagnostic knowledge base. This demonstrates the type of knowledge bases that are useful to aquaculturists. The RIAX allowed the biomass in the system to be doubled (from 40 to 80 fish in a 387 l system) while maintaining acceptable water quality.

A fuzzy logic-based expert system was used to develop an automated denitrifying bioreactor for the removal of nitrate from aquaculture systems (Lee et al., 1995, 1999; Turk et al., 1997; Lea et al., 1998). The operation of the bioreactor (Fig. 2) required the careful control of flow rate (Table 1B) and carbon feed to avoid the production of toxic substances, nitrite and hydrogen sulfide. Manually operated denitrification systems have a history of crashing due to human operator error; the automated bioreactor has the advantage that its operation was monitored 24 h a day by the computer control system. The fuzzy logic-based expert system was validated by direct comparison with the decisions that human experts would have made under similar situations (Whitson et al., 1993; Lee et al., 1999). Fuzzy logic control systems have an advantage over routine set point process control systems because they generate changes in proportion to the degree to which the process deviates from the control range (Table 1B, Lee et al., 1999).

An ANFIS system has been used to implement an aquaculture machine vision system (Whitsell et al., 1997). The objectives for designing the system were to monitor animal activity, size and current condition in situ without disturbing the animals’ behavior. Changing light conditions and turbidity as well as animal movement makes object identification and classification difficult in aquaculture tanks. To overcome these difficulties, the authors used three different DIWs, one to identify animal versus non-animal, a second to estimate near-, mid or far-field and a third to set values for low and high thresholds. The final output of the machine vision system was a histogram of animal movement with a weighted average
regression of activity versus time (Fig. 6); this figure is an example of a report generated using the NetDDE module in the process control software. The main innovation associated with this machine vision system was the use of a training data set based on expert knowledge. The authors found that the expert-modified ANFIS system gave the best performance when compared to earlier systems (Whitsell and Lee, 1994; Whitsell et al., 1997).

In summary, the use of fuzzy control logic appears to offer the best potential to aquaculture because its execution most closely mimics the routine management decisions made by an aquaculturist who is part scientist and artist.

5. Software purchase criteria

The selection of software, whether it is general process control or artificial intelligence software, almost always comes down to some combination of perceived price performance and trust in the developer.
5.1. Functionality/intuitiveness

You should evaluate whether the potential software can accomplish the bottom line goals that you have set for your automated control system. These should be production or economic goals not engineering goals because the price performance of the software system should bolster your financial plan not blow a hole in it. In addition to meeting your goals, the software should not require an excessive time for the manager and staff to learn. It may require some higher level computer skills to install and set-up, but the day-to-day use should be intuitive in the sense that a user should be able to navigate through the screen displays and find the information that they need. The most effective way to evaluate the functionality of the product is to acquire a demonstration copy from the vendor and see if it can perform the required functions. You should verify all control functions, displays, alarms and messages, security procedures, DDE links and mathematical algorithms with valid and invalid data inputs before purchasing (Walters, 1994). Evidence of good documentation in manuals is another sign that the software will be functional and that any problems will be easy to solve. Other good methods to access functionality is to: (1) determine the extent to which the product is used in your industry; (2) discuss its functionality with current users; and (3) evaluate the vendor’s experience in solving the kinds of problems that will be critical for your control system. Finally, you should evaluate the disaster recovery procedures, including backups and recovery.

5.2. Modularity/flexibility

Modularity provides the benefit that system components can be added, moved or deleted as necessary without jeopardizing the integrity of the whole control system (Wolske, 1989; Walters, 1994). This is important for both the software and the hardware. The benefits for modularity in software are that you will not have to buy unnecessary features and that you can upgrade features later. With hardware, the selection of modular communication devices, transmitters, meters, sensors and actuators means that a system can be implemented quickly and modified without major time delays. Modular design also means that once one aquaculture tank monitoring and control system is designed and implemented in your facility, it is easily replicated for all other tank systems.

5.3. Compatibility

Compatibility or connectivity is the measure of how a software product will function with the myriad other software and hardware choices that you will be making in designing and implementing your control system. In terms of hardware compatibility, it means your software should offer a wide range of I/O drivers that are simple to implement (Wolske, 1989). The more proprietary the hardware components are, the more you will pay due to less market competition. This goes for communication standards as well. In terms of software compatibility, it means
using software that contains most of today’s enhancements (e.g. direct dynamic exchange, DDE; object linking and embedding, OLE; shared libraries or dynamic link libraries, DLL; object-oriented programming, OOP; wizards and multimedia systems). If the software is a proprietary product designed for only one type of hardware, you will undoubtedly encounter a problem with interfacing a second party sensor or actuator. Selection of compatible software will ensure that if and when you implement new enhancements (e.g. statistical process control, remote nodes or artificial intelligence) you will be able to do so without buying a completely new system (Wolske, 1989).

5.4. Upgrade path/service

This is one of the most critical aspects for evaluating a software program, yet it is the hardest to predict because its effects will be felt at some point in the future. One of the salient characteristics of computer technology is that nothing will remain static for very long. In practical terms, this means that the process control software will continue to improve and it will be more and more demanding on system resources, requiring software and hardware upgrades (Labs, 1992). The best recent example of this has been the migration of PC users from Microsoft’s DOS™ to Windows™ to Windows for Workgroups™ to Windows 95 or 98™ to Windows NT™. Each step required new software (both applications as well as operating system) and improved hardware (more memory, larger disk drives, faster CD-ROMs and better graphics cards). Each user has to judge the price performance benefits of each potential upgrade step (Huber, 1994). This can best be evaluated by asking the vendor specific questions concerning the past upgrade history and pricing policy as well as questions concerning planned improvements to the software. There are three things that you should consider: (1) you do not have to upgrade, it’s your decision; (2) if you buy from stable vendors, your applications will work well with whichever operating system it is designed to use; and (3) your process control system will become so valuable to you that you will be seeking to improve its productivity constantly (Whitson et al., 1997). When you finally select the product, remember that the vendor is becoming a partner in your business so select a vendor that is committed to your success (Walters, 1994).

5.5. Hardware requirements

The choice of process control software will largely determine the hardware platform and the resources needed for efficient operation (Lee, 1995). Generally, the computer and electronic hardware will be less expensive than the software unless a very large array of I/O channels is necessary. Once the software has been chosen, a computer with adequate resources (e.g. memory, graphics card, disk space, backup system, and multimedia) should be selected. The next decision involves the choice of an appropriate I/O system and I/O drivers. Most widely distributed process control software programs will have I/O driver options for most of the industry standard I/O systems.
5.6. **Price performance**

Finally, all the above criteria must be balanced against the cost of purchase for the software. This decision should be made based on a ‘Top Down’ approach, meaning that the selected system must meet your economic goals not just make the process easier to manage (Christie, 1989; Lee, 1995). While the latter may be important to your employees, the real business reason for considering a process control system is to increase production, reduce labor and material costs and/or reduce waste.

**Acknowledgements**

This research was supported by grants from the Texas A&M University Sea Grant Program (R/M-58), National Sea Grant College Program (R/M-53), the Small Business Innovative Research Program of the Department of Commerce and The Texas Higher Education Coordinating Board’s Advanced Technology (004952-068) and Technology Development and Transfer (004952-079) programs and an educational partnership with Intellution, Inc., Norwood, MA as well as the Marine Medicine budget of the Marine Biomedical Institute, University of Texas Medical Branch.

**References**


Wolske, B.K., 1989. Selecting the right software. Control 11 (9), 50–64.