Artificial intelligence tools can aid sensor systems

At least seven artificial intelligence (AI) tools can be useful when applied to sensor systems: knowledge-based systems, fuzzy logic, automatic knowledge acquisition, neural networks, genetic algorithms, case-based reasoning, and ambient intelligence.

Seven artificial intelligence (AI) tools—knowledge-based systems, fuzzy logic, automatic knowledge acquisition, neural networks, genetic algorithms, case-based reasoning, and ambient intelligence—are reviewed that have proved to be useful with sensor systems. They are: knowledge-based systems, fuzzy logic, automatic knowledge acquisition, neural networks, genetic algorithms, case-based reasoning, and ambient intelligence. Each AI tool is outlined, together with some examples of its use with sensor systems. Applications of these tools within sensor systems have become more widespread due to the power and affordability of present-day computers. Many new sensor applications may emerge, and greater use may be made of hybrid tools that combine the strengths of two or more of the tools reviewed.

The tools and methods reviewed here have minimal computation complexity and can be implemented with small sensor systems, single sensors, or sensor arrays with low-capability microcontrollers. The appropriate deployment of the new AI tools will contribute to the creation of more competitive sensor systems and applications. Other technological developments in AI that will impact sensor systems include data mining techniques, multi-agent systems, and distributed self-organizing systems. Ambient sensing involves integrating many microelectronic processors and sensors into everyday objects to make them "smart." They can explore their environment, communicate with other smart things, and interact with humans. Advice provided aims to help users cope with their tasks in intuitive ways, but the repercussion of such integration into our lives is difficult to predict. Using ambient intelligence and a mix of AI tools uses the best of each technology. The concepts are generically applicable across industrial processes. Research below intends to show these concepts can work in practice.

Creating smarter sensor systems

Sensor systems can be improved using artificial intelligence (AI). AI emerged as a computer science discipline in the mid 1930s, and it has produced a number of powerful tools that are useful to sensor systems for automatically solving problems that would normally require human intelligence. Seven such tools are here: knowledge-based systems, fuzzy logic, inductive learning, neural networks, genetic algorithms, case-based reasoning, and ambient intelligence.

AI systems have been improving, and new advances in machine intelligence are creating seamless interactions between people and digital sensor systems. Although the introduction of AI into the industry has been slow, it promises to bring improvements in flexibility, reconfigurability, and reliability. New machine systems are exceeding human performance in increasing numbers of tasks. As they merge with us more intimately, and we combine our brain power with computer capacity to deliberate, analyze, deduce, communicate, and invent, then we may be on the threshold of a new age of machine intelligence.

AI (or machine intelligence) combines a wide variety of advanced technologies to give machines an ability to learn, adapt, make decisions, and display new behaviors. This is achieved using technologies such as neural networks, expert systems, self-organizing maps, fuzzy logic, and genetic algorithms, and that machine intelligence technology has been developed through its application to many areas where sensor information has needed to be interpreted and processed, for example, assembly, biosensors, building modeling, computer vision, cutting tool diagnosis, environmental engineering, force sensing, health monitoring, human-computer interaction, Internet use, laser milling, maintenance and inspection, powered assistance.
Figure 2 shows an architecture for a fuzzy logic-based controller.

Figure 3 shows an experimental system to test the use of ambient intelligence to improve energy efficiency.

Figure 4, a proposed system flow diagram shows how a system could gather data from an image sensor. Visual data and CAD model data will be used in conjunction to determine an object list; that object list will be passed to a weld identifier module that will use AI techniques to determine weld requirements.

Robotics, sensor networks, and teleoperation.

These developments in machine intelligence are being introduced into ever more complex sensor systems. The click of a mouse, the flick of a switch, or the thought of a brain might convert almost any sensor data to information and transport it to you. Recent examples of this research work are provided, which include work at the University of Portsmouth. Seven areas follow where AI can help sensor systems.

1. Knowledge-based systems

Knowledge-based (or expert) systems are computer programs embodying knowledge about a domain for solving problems related to that domain. An expert system usually has two main elements, a knowledge base and an inference mechanism. The knowledge base contains domain knowledge which may be expressed as a combination of “IF-THEN” rules, factual statements, frames, objects, procedures, and cases. An inference mechanism manipulates stored knowledge to produce solutions to problems. Knowledge manipulation methods include using inheritance and constraints (in a frame-based or object-oriented expert system), retrieval and adaptation of case examples (in case-based systems), and the application of inference rules (in rule-based systems), according to some control procedure (forward or backward chaining) and search strategy (depth or breadth first).

A rule-based system describes knowledge of a system in terms of IF... THEN... ELSE. Specific knowledge can be used to make decisions. These systems are good at representing knowledge and decisions in a way that is understandable to humans. Due to the rigid rule-base structure they are less good at handling uncertainty and are poor at handling imprecision. A typical rule-based system has four basic components: a list of rules or rule base (a specific type of knowledge base); an inference engine or semantic reasoner (infers information or takes action based on the interaction of input and the rule base); temporary working memory; and a user interface or other connection to the outside world through which input and output signals are received and sent.

The concept in case-based reasoning is to adapt solutions from previous problems to current problems. These solutions are stored in a database and can represent the experience of human specialists. When a problem occurs that a system has not experienced, it compares with previous cases and selects one that is closest to the current problem. It then acts upon the solution given and updates the database depending upon the success or failure of the action. Case-based reasoning systems are often considered to be an extension of rule-based systems. They are good at representing knowledge in a way that is clear to humans, and have the ability to learn from past examples by generating new cases.

2. Case-based reasoning

Case-based reasoning has been formalized for purposes of computer reasoning as a four-step process: 1) Retrieve: Given a target problem, retrieve cases from memory that are relevant to solving it. A case consists of a problem, its solution, and, typically, annotations about how the solution was derived.

2) Reuse: Map the solution from the previous case to the target problem. This may involve adapting as needed to fit the new situation.

3) Revise: Having mapped the previous solution to the target situation, test the new solution in the real world (or a simulation) and, if necessary, revise.

4) Retain: After the solution has been suc-
cessfully adapted to the target problem, store the resulting experience as a new case in memory.

Critics argue that it is an approach that accepts anecdotal evidence as its main operating principle. Without statistically relevant data for backing and implicit generalization, there is no guarantee that the generalization is correct. All inductive reasoning where data is too scarce for statistical relevance is based on anecdotal evidence.

The concept in case-based reasoning (CBR) is to adapt solutions from previous problems to current problems. These solutions are stored in a database and represent the experience of human specialists. When a problem occurs that a system has not experienced, it compares with previous cases and selects one closest to the current problem. It then acts upon the solution given and updates the database depending upon the success or failure of the action.

CBR systems are often considered to be an extension of rule-based systems. As with rule-based systems, CBR systems are good at representing knowledge in a way clear to humans; however, CBR systems also have the ability to learn from past examples by generating additional new cases. Figure 1 shows a CBR system.

Many expert systems are developed using programs known as “shells,” which are ready-made expert systems complete with inferencing and knowledge storage facilities but without the domain knowledge. Some sophisticated expert systems are constructed with help of “development environments.” The latter are more flexible than shells, providing means for users to implement their own inferencing and knowledge representation methods.

Expert systems are probably the most mature among tools here, with many commercial shells and development tools available to facilitate their construction. Once the domain knowledge to be incorporated in an expert system has been extracted, the process of building the system is relatively simple. The ease with which expert systems can be developed has led to a large number of applications of the tool. In sensor systems, applications can be found for a variety of tasks, including selection of sensor inputs, interpreting signals, condition monitoring, fault diagnosis, machine and process control, machine design, process planning, production scheduling, and system configuring. Specific tasks undertaken by expert systems are assembly, automatic programming, controlling intelligent complex vehicles, planning inspection, predicting risk of disease, selecting tools and machining strategies, sequence planning, and controlling plant growth.

3. Fuzzy logic

A disadvantage of ordinary rule-based expert systems is that they cannot handle new situations not covered explicitly in their knowledge bases (that is, situations not fitting exactly those described in the "IF" parts of the rules). These rule-based systems are unable to produce conclusions when such situations are encountered. They are shallow systems which fail in a "brittle" manner, rather than exhibit a gradual reduction in performance when faced with increasingly unfamiliar problems, as human experts would.

The use of fuzzy logic, which reflects the qualitative and inexact nature of human reasoning, can enable expert systems to be more resilient. With fuzzy logic, the precise value of a variable is replaced by a linguistic description, the meaning of which is represented by a fuzzy set, and inferencing is carried out based on this representation. For example, an input from a sensor system of 20 can be replaced by "normal" as the linguistic description of the variable "sensor input." A fuzzy set defining the term "normal sensor input" might be: normal sensor input = 0.0/below 10 widgets per minute + 0.5/10–15 widgets per minute + 1.0/15–25 widgets per minute + 0.5/25–30 widgets per minute + 0.0/above 30 widgets per minute.

The values 0.0, 0.5, and 1.0 are the degrees or grades of membership of the sensor ranges below 10 (above 30), 10–15 (25–30), and 15–25 to the given fuzzy set. A grade of membership equal to 1 indicates full membership, and a null grade of membership corresponds to total non-membership.

Expert system knowledge employing fuzzy logic can be expressed as qualitative statements (or fuzzy rules), such as, "If the input from the room temperature sensor is normal, then set the heat input to normal." A reasoning procedure known as the compositional rule of inference (the equivalent of the modus-ponens rule in rule-based expert systems) enables conclusions to be drawn by generalization (extrapolation or interpolation) from the qualitative information stored in the knowledge base. When a sensor input is detected to be "slightly below normal," a controlling fuzzy expert system might deduce that the sensor inputs should be set to "slightly above normal." (This conclusion might not have been contained in any fuzzy rule stored in the system.)
Fuzzy expert systems (FES) use fuzzy logic to handle the uncertainties generated by incomplete or partially corrupt data. The technique uses the mathematical theory of fuzzy sets to simulate human reasoning. Humans can easily deal with ambiguity (areas of gray) when decision making, yet machines find it difficult. Figure 2 shows an architecture for a fuzzy logic-based controller.

Fuzzy logic has many applications in sensor systems where the domain knowledge can be imprecise. Fuzzy logic is well suited where imprecision is inherent due to imprecise limits between structures or objects, limited resolution, numerical reconstruction methods, and image filtering. Applications in structural object recognition and scene interpretation have been developed using fuzzy sets within expert systems. Fuzzy expert systems are suitable for applications that require an ability to handle uncertain and imprecise situations. They do not have the ability to learn as the values within the system are preset and cannot be changed.

Fuzzy logic successes have been achieved in the areas of cooperative robots, mobile robots, prediction of sensory properties, supply chain management, and welding.

4. Automatic knowledge acquisition

Getting domain knowledge to build into a knowledge base can be complex and time consuming. It can be a bottleneck in constructing an expert system. Automatic knowledge acquisition techniques were developed to address this, for example, in the form of IF THEN rules (or an equivalent decision tree). This sort of learning program usually requires a set of examples as a learning input. Each example is characterized by the values of a number of attributes and the class to which they belong. One approach is through a process of "dividing-and-conquering," where attributes are selected according to some strategy (for example, to maximize the information gain) to divide the original example set into subsets, and the inductive learning program builds a decision tree that correctly classifies the given example set. The tree represents the knowledge generalized from the specific examples in the set.

In another approach, called the "covering approach," the inductive learning program attempts to find groups of attributes uniquely shared by examples in given classes and forms rules with the IF part as conjunctions of those attributes and the THEN part as the classes. The program removes correctly classified examples from consideration and stops when rules have been formed to classify examples in the set.

Another approach is to use logic programming instead of propositional logic to describe examples and represent new concepts. This approach employs the more powerful predicate logic to represent training examples and background knowledge and to express new concepts. Predicate logic permits the use of different forms of training examples and background knowledge. It enables the outputs of the induction process (the induced concept) to be described as general first-order clauses with variables and not just as zero-order propositional clauses made up of attribute-value pairs. There are two main types of these systems, the first based on the top-down generalization-specialization method, and the second on the principle of inverse resolution.

A number of learning programs have been developed, for example ID3, which is a divide-and-conquer program; the AQ program, which follows the covering approach; the FOIL program, which is an ILP system adopting the generalization-specialization method; and the GOLEM program, which is an ILP system based on inverse resolution. Although most programs only generate crisp decision rules, algorithms also have been developed to produce fuzzy rules.

The requirement for a set of examples in a rigid format (with known attributes and of known classes) has been easily satisfied by requirements in sensor systems and networks so that automatic learning has been widely used in sensor systems. This sort of learning is most suitable for problems where attributes have discrete or symbolic values rather than those with continuous-valued attributes as in many sensor systems problems.

Examples of inductive learning applications are laser cutting, mine detection, and robotics.

5. Neural networks

Neural networks can also capture domain knowledge from examples. They do not archive the acquired knowledge in an explicit form such as rules or decision trees, and they can readily handle continuous and discrete data. They also have a good generalization capability, as with fuzzy systems. A neural network is a computational model of the brain. Neural network models usually assume that computation is distributed over several simple units called neurons, which are interconnected and operate in parallel. (Neural networks also are called parallel-distributed-processing systems or connectionist systems.)

The most popular neural network is the multilayer perceptron, which is a feedforward network; all signals flow in one direction from the input to the output of the network. Feedforward networks can perform static mapping between an input space and an output space: the output at a
Many new sensor systems applications may emerge, and greater use may be made of hybrid tools that combine the strengths of two or more of these tools.

given instant is a function only of the input at that instant. Recurrent networks, where the outputs of some neurons are fed back to the same neurons or to neurons in layers before them, are said to have a dynamic memory: the output of such networks at a given instant reflects the current input as well as previous inputs and outputs.

Implicit "knowledge" is built into a neural network by training it. Some neural networks can be trained by being provided with typical input patterns and the corresponding expected output patterns. Error between the actual and expected outputs is used to modify the strengths, or weights, of the connections between the neurons. This method is known as supervised training. In a multi-layer perceptron, the back-propagation algorithm for supervised training is often adopted to propagate the error from the output neurons and compute the weight modifications for the neurons in the hidden layers.

Some neural networks are trained in a supervised mode, where only input patterns are provided in training and the network learns automatically to make groups with similar features.

Artificial neural networks (ANNs) typically have inputs and outputs, with processing within hidden layers in between. Inputs are independent variables and outputs are dependent. ANNs are flexible mathematical functions with configurable internal parameters. To accurately represent complicated relationships, these parameters are adjusted through a learning algorithm. In "supervised" learning, examples of inputs and corresponding desired outputs are simultaneously presented to networks, which iteratively self-adjust to accurately represent as many examples as possible. Once trained, ANNs can accept new inputs and attempt to predict accurate outputs. To produce an output, the network simply performs function evaluation. The only assumption is that there exists some continuous functional relationship between input and output data. Neural networks can be employed as mapping devices, pattern classifiers, or pattern completers (auto-associative content addressable memories and pattern associators).

Recent applications are feature recognition, heat exchangers, inspection of soldering joints, optimizing spot welding parameters, power, tactile displays, and vehicle sensor systems.

7. Ambient intelligence

Ambient intelligence has been promoted for the last decade as a vision of people working easily in digitally controlled environments in which the electronics can anticipate their behavior and respond to their presence. The concept of ambient intelligence is for seamless interaction between people and sensor systems to meet actual and anticipated needs. Use in industry has been limited, but new, more intelligent and more interactive systems are at the research stage.

Expanding systems, intelligence

AI can increase effective communication, reduce mistakes, minimize errors, and extend sensor life. Over the past 40 years, artificial intelligence has produced a number of powerful tools, including those here. Use of these tools in sensor systems applications have become more widespread due to the power and affordability of present-day computers. Many new sensor systems applications may emerge, and greater use may be made of hybrid tools that combine the strengths of two or more of these tools. Other developments in AI that will impact sensor systems include data mining, multi-agent systems, and distributed self-organizing systems. Appropriate deployment of new AI tools will contribute to more competitive sensor systems. It may take another decade for engineers to recognize the benefits given the current lack of familiarity and the technical barriers associated with using these tools. The field of study is expanding. Tools and methods have minimal computation complexity and can be implemented on small assembly lines or single robots, or systems with low-capability microcontrollers. These approaches proposed use ambient intelligence and the mixing of different AI tools in an effort to use the best of each across many processes.