Plant Operations: Thoughts on Intelligent Control

W.J. BUCHANAN and W.N. PEARSON
School of Computing, Napier University, Edinburgh, UK

1 Introduction

Industrial plants have, in the past, been controlled by human operators, aided by computer controlled equipment. At present, the computer aids the control of the plant, but humans still monitor its operation and make changes to the control program to meet the objectives of the plant. Human operators and managers also define production targets, optimisation levels, set maintenance levels, deploy technical staff, organise plant shutdowns, specify equipment, and so on. These tasks are seen as high-level tasks to the actual control of the plant, which, once defined, is a dependable task which has a great deal of repetition. Computers cope with repetition much better than humans, who like to do a variety of things.

This method of operation, where the computer controls and the human is left to organise the high-level tasks is likely to change as intelligent computers take over these tasks. At present, the computer controls the industrial plant in a mechanistic way, where a human programmer codes the control software with:

- Conditional programming. A series of if...then...else statements.
- Event-driven programming. The computer reacts to events from the plant, such as changes in speed, pressures, and so on.
- Emergency procedures. Not every condition and event can be programmed for and programmers often have emergency (or exception) procedures for an unknown conditions or events. This might be simply to query the operator for some input, or to shutdown the plant.

These programming techniques normally require:

- Mathematical models of the plant.
- Plant simulations, which are used to optimise processes.
- Discussions with expert operators (to create an expert system).

These are good methods, but what happens when:

- The mathematical model is not accurate.
- The plant simulation does not model the actual operation of the plant.
- The plant changes its operation over time.
- The expert isn’t the best expert, or did not give the full information on the operation of the plant.

A computer will blindly depend on the programmer to control the equipment, and it is not allowed to change its own operation depending on plant operations. The most important thing is the computer taking over proper control of the plant is for the computer to know the overall objectives of the plant. This is illustrated in Figure 1.

The objectives for the plant might one or a mixture of the following:

- To maximise plant safety.
- To minimise plant operating costs.
- To minimise plant emissions.
- To maximise product quality.
- To minimise plant downtime.
- and so on.
It is likely that the plant has not just one objective, but many, each with its own level of importance. For example, safety is always the most important objective, which most humans understand, but a computer must be programmed to achieve this. Then, once the plant is safe, the plant must meet its legal and moral obligations, such as for the amount of emissions, product quality, and so on. After this, a major object is to optimise the plant and maximum profits. It is thus a complex problem, and humans will always err on the safety, reducing downtimes, and so on.

2 Computers and Artificial Intelligence

Computers and humans each have advantages over each other, but as computers become faster and contain more memory, they can replace humans in many situations. In Arthur C. Clarke’s 2001: A Space Odyssey, the spacecraft’s on-board computer, HAL, played the captain. The computer won and then took over the ship. Thus if a computer could beat the best human intellect at a game which provided one of the greatest human challenges then they are capable of taking on the most complex of problems. This because a reality when, on 10th February 1996, Deep Blue, a computer developed by IBM, beat Gary Kasparov at chess in a match in the USA [1]. It was a triumph of Artificial Intelligence (AI) over human intelligence. Gary actually went on to beat Deep Blue by four games to two, but the damage had already been done. It would only be a matter of time before a computer would beat the chess champion, as, on average, they increase their processing capacity by 50% each, as well improving their operation and the amount of information they can store. Thus, in a rematch in May 1997, Deeper Blue, the big brother of Deep Blue, beat Gary by 3½ to 2½. The best computer had finally beaten the best human brain. In reality, it was unfair challenge. Computers have a massive number of openings programmed into it. Kasparov knew the only have to beat the computer was to get it away from its opening encyclopaedia as quickly as possible, he thus made moves which would be perceived as bad when playing against another human. Computers also process data faster than the human brain and can search billions of different options to find the best.

Claude Shannon, in 1949, listed the levels in which computers could operate, each with a higher level of intellectual operation. Table 1 outlines these.

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Machines for designing filters, equalisers, and so on.</td>
<td>CAD computers have been available for some time to design electrical and electronic circuits. These tend to be based on mathematical calculations.</td>
</tr>
<tr>
<td>2</td>
<td>Machines for designing relay and switching circuits.</td>
<td>Digital logic design software packages have also been available for some time. These can design using high-level languages, such as VHDL or from hierarchical schematics. They tend to be based on Boolean logic.</td>
</tr>
</tbody>
</table>
3 Machines which will handle the routing of telephone calls based on individual circumstances rather than fixed patterns.

Switching exchanges have been available for some time, which route calls depending on the called number. Modern computers can analyse the data being sent and finds the best route for the type of data.

4 Machines for performing symbolic (non-numeric) mathematical operations.

Computers generally operate on data which is non-numeric, such as database applications or image processing. Neural networks are currently used for predicting the weather.

5 Machines capable of translating from one language to another.

Computers now act as translator for one language to another.

6 Machines for making strategic decisions in simplifying military operations.

Computers are beginning to make strategic decisions, but these are still checked by humans.

7 Machine capable of orchestrating a melody.

As yet computers cannot properly create music, which is pleasing to the human ear. They can generally mimic musical style, but things that make music pleasant to the ear is difficult to define.

8 Machines capable of logical deduction.

As yet, there are few practical applications in this area.

---

Table 2 outlines some of the main characteristics between computers and humans.

<table>
<thead>
<tr>
<th></th>
<th>Computers</th>
<th>Humans</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
<td>Faster</td>
<td>Humans</td>
<td>A human brain takes several milliseconds to sense changes and even longer to react to them. Fast computers can react and react to changes with millionth of a second.</td>
</tr>
<tr>
<td>Predictability</td>
<td>Better</td>
<td></td>
<td>This is because they react to events in a predictable way, which is dictated by their computer program.</td>
</tr>
<tr>
<td>Stamina</td>
<td>Better</td>
<td></td>
<td>Computers will generally work 24-hours a day, 7 days a week and 365 days a year. They also do not stop for lunch.</td>
</tr>
<tr>
<td>Learning</td>
<td>Better</td>
<td></td>
<td>Computers normally run a fixed program, and which cannot change when the controlled system changes. Humans learn from their mistakes and easy adapt to varying conditions.</td>
</tr>
<tr>
<td>Reliability</td>
<td>Better, sometimes</td>
<td></td>
<td>Computers, and their programs, need to be maintained and fixed when they have problems. As most computers cannot learn from their mistakes, they cannot fix themselves. Also most computers do not actually know what their main target is, and how to achieve it (in most cases it is safety, regulations and then financial).</td>
</tr>
<tr>
<td>Loyalty</td>
<td>Totally</td>
<td></td>
<td>Computers, like dogs, are loyal servers. They have no interest in moving to other jobs, or demand better working conditions.</td>
</tr>
<tr>
<td>Strategy</td>
<td>Better</td>
<td></td>
<td>Humans have a great knack of developing long-term strategies from complex situations. For example, in a chess program, a computer will generally search many combinations to find the best one, whereas a human will play with a strategy, which is ever changing.</td>
</tr>
<tr>
<td>Enterprise</td>
<td>Best</td>
<td></td>
<td>Most new ventures are initiated by humans. It is difficult to program computers to create enterprise, as it is a complex subject with many inputs, which vary over time, and can have many results.</td>
</tr>
<tr>
<td>Creative</td>
<td>Best</td>
<td></td>
<td>Humans have different motivations than computes. Humans are driven by many different thinks, such as love, beauty, money, hate, spite, and so on. These make humans both predictable and unpredictable. It is extremely difficult to program computers to be creative.</td>
</tr>
</tbody>
</table>
It can be seen that humans have two great advantages over computers. These are:

- **Learning.** Humans adapt to changing situations, and generally quickly learn tasks. Unfortunately, once these tasks have been learnt, they often lead to boredom if they are repeated repetitively.
- **Strategy.** Humans are excellent at taking complex tasks and splitting them into smaller tasks. Then, knowing the outcome, they can implement these in the required way, but can make changes depending on conditions.
- **Enterprise.** Computers, as they are programmed at the present, are an excellent business tool. They generally allow better decision making, but, at present, they cannot initiate new events.
- **Creative.** As with enterprise, humans are generally more creative than computers. This will change over the coming years as they are programmed with the aid of psychologists, musicians and artists, and will contain elements which are pleasing to the human senses.

The key to improved strategy, increased enterprise and increased creativity is: LEARNING. If computers could learn, especially from their mistakes, they could outperform humans in most tasks. The key to learning is the neural network computer, which involves a computer thinking in the same way as a human does. Neural networks are based on the study of neural processing in the brain. Each of the processing elements with a neural network is similar to the neurons in a human brain, and hence, they are referred to as neurons, or artificial neurons, often known simply as neurons.

### 3 Feed-forward neural networks

Neural networks contain neurons which are connected by synapses. Figure 1.1 shows a layered feed-forward neural network, which contains neurons and the connection between neurons represented by edges of a directed graph. Circular nodes represent neurons.

A layered feed-forward neural network contains processing elements which are build into layers, or subgroups. Data propagates from the input layer, to the next, and finally to the output layer. The layers between the input and output layers are known as hidden layers. Each layer independently operates on the data and passes it on to the next. The processing elements compute with the following:

- Weighted sum of the inputs.
- The possible application of a threshold function, which qualifies the output of a neuron in the output layer.

This example network in Figure 2 has three layers: an input layer, a hidden layer and an output layer.

![Figure 2 Feed-forward neural network](image)

### 4 Neural network concepts

The basic concepts of neural networks are:

- **Neuron output.** A neuron’s output is the weighted sum of its inputs. A threshold value then be applied to determine the final value. A neuron has ‘fired’ if its output value is a ‘1’, else, if its output is a zero then it is not ‘fired’.
Weights. These define how weightings of the connections between different layers and characterize the network. To set weightings, either:

- No training. The network is run with a fixed set of weightings.
- Training. The network starts with an initial set of weightings and then these are modified when the network is run. This continues until the final goal is achieved. This allows a network to learn. This learning can either be supervised or unsupervised.

Feedback. An important concept in learning is feedback, where the output is fed back to the input so that the error can be found. The weightings in the network can then be adjusted to try and improve the error. This feedback can be to the input layer or any layer between the input layer and the output layer, sometimes labelled the hidden layer. Feedback continues until the network finds the best set of weightings.

5 Learning

Just like humans, a neural can learn either supervised or unsupervised. Supervised learning involves using an external criteria, which is used to match the output of the network. An unsupervised network is also known as self-organising and there is generally more interaction between neurons, typically with feedback and intralayer connections between neurons.

An example of a supervised network is arranging bank loans. The network would be presented with bank loan details (the stimulus), along with how well the history of payments of previous loans (the metrics). The training would then involve the prediction on whether a borrower would be likely to default on a loan. An unsupervised network only requires a stimulus. In the bank loan example, the network would initially get bank loan details and would learn from its own mistakes.

It is important for a neural network to remember how it learned from training data. Thus the network requires a long-term memory (LTM), as well as a short-term memory (STM). Long-term memory must decays over a given time interval, otherwise the network becomes too burdened with data that occurred a long time ago. This effects the recall of the network.

5.1 Noise

Noise is any unwanted deviation from the actual data. In an electronic system noise is the electrical energy which is added to a signal. This noise reduces the quality of the signal and if the noise is too great, it may swamp the signal. In a neural network, the training data may have some noise in it. For example, in the bank loan example, a borrower may default on a loan, but has an excellent credit history. It is obviously important for the neural network to cope with noise. After training the network, should stabilise on meaningful values. This is called network convergence. It does no good for the network to give different values for the same set of input data. After convergence, the network should be tested to see how it copes with intentionally added noise. The network should still be able to converge with the noisy data.

6 Neural network distinction

Three important decisions are made in the design and development of a neural network. These are:

- Structure. This defines the architecture and topology of the network, such as how many layers it has, their functionality and the interconnections between neurons.
- Encoding. This defines the method of changing the weights. In a multilayer feed-forward neural network the weights initially have a random value. Then, by training, a backpropagation algorithm is used to update the updating weights, starting from the output backwards. On a feed-forward network, the final values of the weights are then fixed.
- Recall. This defines the method and capacity to retrieve information. It involves getting an expected output for a given input. If the neural network is operating correctly then the same output should result if it is presented consecutively with the same input. Recall can either be autoassociate or heteroassociate. Autoassociation involves associating an input vector with itself as the output, whereas heteroassociation involves recalling an output vector for a given an input vector. For example, a Hopfield neural makes an association between different patterns (heteroassociated) or associates a pattern with itself (autoassociation).

7 Industrial Control

Control systems are often required to deal with complex applications, sometimes having several process variables which require non-linear solutions. Such applications may have been traditionally
solved using skilled operators or complex model-based software, as illustrated in Figure 3. Both solutions are costly, as the former requires operator training and the presence of a skilled operator and the latter requires sophisticated and costly software, and specialist support.

Botros [2], identified that advances in industrial control would be through, principally, “...incremental improvements in existing technology”, such as the implementation of different computational techniques. These could include the application of Neural Networks and Fuzzy Logic. The learning properties of Neural Networks and heuristics of Fuzzy Logic based controllers allows the control of plant where formal knowledge of the plant process may be limited, but where data describing plant performance may be abundant. Typical applications may include:

- Load balancing of energy generation and transmission plant.
- Predicting and controlling plant responses to changes in load demand.
- Selection of optimal plant operating point.

The aim of implementing plant control using AI techniques is to improve business performance [3], this can be through improved process management, such as:

- Improved scheduling.
- Improved planning.
- Increased optimisation.

These allow for more reliable strategic decision-making and better-informed decisions, which should make for better business performance.

8 Generic Plant Control Structure

A hierarchical plant control structure can be represented as high, intermediate and low-level tasks. Each of these tasks can be grouped into five areas:

- Plant management.
- Planning and scheduling.
- Advanced control.
- Conventional control.
- Data acquisition and reconciliation.

These areas are interdependent, that is, an instrument failure could impact on higher activity level tasks such as maintenance planning or production targets. Similarly, managerial decisions may affect operating procedures and plant performance at intermediate or lower levels.

Figure 4 shows a typical hardware hierarchy. The stronger the information chain through these levels, the less the likelihood that plant may run sub-optimally. Hence, each level requires knowledge from the other. In addition, the higher levels require external knowledge such as sales and marketing information.
9 Intelligent Control Functionality

In modern SCADA-based plant control systems there are areas of overlap between functions which were once implemented by discrete controllers. Now a PID function might be implemented using SCADA software. Fuzzy logic-based rules can be implemented in a PLC, as well as in SCADA software modules. Figure 5 shows the overlaps.

It can be seen from Figure 5 that the information becomes available to the operator at the SCADA operator interface. Plant performance is then very much dependent on the operator’s ability to assimilate information from many sources, then manipulate controllers to achieve production targets. As a secondary task the operator may then attempt to adjust the plant to run optimally.

For a given set of operating conditions the operator must have prior knowledge or experience of these conditions to know controlled variable settings for optimal operation of the plant. Figure 5 illustrates the learning or training process which the operator is likely to have gone through.

Differences between the actual and expected performances generate a deviation signal. This signal could be due to non-conformance of the plant behaviour after equipment failure, poor plant modelling or lack of understanding of plant processes. The deviation signal might imply change controller settings or re-training the operator.
10 Knowledge against Data-Driven Control Strategies

Operator training results in a knowledge-based, expert system, that is, the operator. After training the operator is able to make causal connections between observed plant conditions and plant performance, usually in terms of production or yield performance. With experience of plant operation the operator is able to adjust key process parameters to influence or control plant behaviour. The data from the plant, fed into or calculated within the SCADA, can be used to predict or model plant performance as a function of key variables. Modelling can be achieved by applying mathematical laws to the data or using statistical techniques, such as multivariable regression. Highly non-linear and highly interactive processes are very difficult to model, which results in poor accuracy and the predicative qualities of models.

Neural networks are being increasingly used to predict non-linear plant performance, and are trained to implement control functions. The notion of training implies a supervised learning category of Neural Networks.

11 An overview of a Supervised Training Network

Generally a feed-forward network [4]–[6], using a back-propagation learning algorithm is used for control type applications. The Neural Network is trained on a data set which consists of an input data set and corresponding training targets. The network is presented with an input data set and its corresponding output is compared with the training target. The difference between the actual output and the target is reduced using back propagation. This technique minimises the error between the current layer and preceding layer of the network in successive steps until the input layer is reached.
A feed-forward, back-propagation trained network is also referred to as a Multi-Layered Perceptron, MLP [7]. More recently the use of clustered Radial Basis Function, RBF (for example, Gaussian) networks have been proposed.

12 Generic Application of Neural Networks for Plant Control

Current control applications of neural networks are restricted to advisory type roles where they may predict plant operating/set-point but leave the final decision to operator discretion [8,9]. This might be represented on the technology diagram as shown in Figure 7.

A common implementation of Neural Network control is to use two networks, one to emulate the forward process and one to learn the reverse process. Plant inputs are fed into the emulator network and the network is trained to model selected plant parameters or performance descriptors. The reverse process network is trained using the performance descriptor output by the emulator to learn key set points which will reproduce the performance descriptors modelled by the emulator network. This type of architecture is shown in Figure 8, which allows the networks to be trained on-line. An adaptive controller based on Neural Network non-linear tuning of controller constants is shown in Figure 9 [10].

One of the most recent developments in Neural Network control is the tracking Neuro-controller, which tracks the plant operating point over time, as illustrated in Figure 10. The goodness factor is measure of the closeness of the set point to the optimal point [11].
13 Conclusion

This paper has presented some of the most widely used techniques in large-scale industrial control using Neural Networks. A major limiting factor in using Neural Network is the heavy computing power required and their speed of response in real-time control. The RBF technique has great potential overcoming these limitations as it can be trained over a smaller area of overall plant operation. This reduces speed of response and reduced computing power, because, in general the data sets are smaller and more manageable. Current research, which is commercially sensitive, is investigating the control of large-scale industrial plant.

14 References