ISSN No. 0976-5697

Volume 2, No. 4, July-August 2011

International Journal of Advanced Research in Computer Science

TECHNICAL NOTE

Available Online at www.ijarcs.info

# Intelligent Control In Industries

Nishant Dubey
School of Computer & Electronics
IPS Acdemy, Indore, India
nishantdubey@sify.com

***Abstract:*** Intelligent control takes a radically different approach to the control of industrial processes and plants from conventional control. The knowledge and experience of human operators constitutes the basis for this new approach to control engineering for which computational intelligence provides the theoretical foundation. Traditionally, intelligent control has embraced classical control theory, neural networks, fuzzy logic, classical AI, and a wide variety of search techniques (such as genetic algorithms and others).In this paper we summarize the potential and some limitation of intelligent control and we attempt to address the questions on how, where, when and under what conditions can intelligent control be applied in practice.

***Keywords:*** Fuzzy System, AI, Control, Machine, Operator , Logic, Process, Plant

## I. INTRODUCTION

Intelligent control seeks solution to the problem of controlling plants from the viewpoint of the human operator. In other words the technique seeks to establish some kind of cognitive model of the human operator and not the plant under his control. This is the point at which intelligent control depart from conventional control and it is undoubtedly true that the technique could not have been possible but for the rapid process in computer technology. Computational intelligence provides the tools with which to make intelligent control a reality. The reproduction of human intelligence and the mechanism for inferring decisions on the appropriate control actions, strategy or policy that must be followed are embedded in these tools.

## II. INTELLIGENT CONTROL

Intelligent control is the use of general-purpose control systems, which learn over time how to achieve goals (or optimize) in complex, noisy, nonlinear environments whose dynamics must ultimately be learned in real time. This kind of control cannot be achieved by simple, incremental improvements over existing approaches.

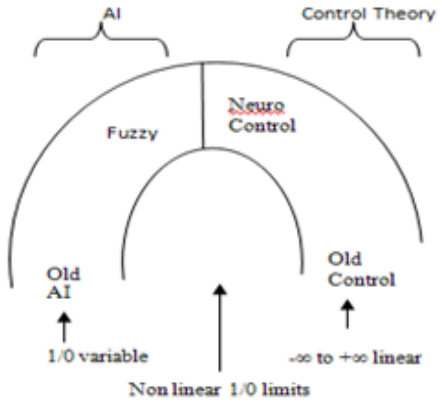


Figure 1: Aspects of intelligent control

Figure1 illustrates more generally our view of the relations between control theory, neurocontrol, fuzzy logic, and AI. Just as neurocontrol is an innovative subset of control theory, so too is fuzzy logic an innovative subset of AI. (Some other parts of AI belong in the upper middle part of Figure1 as well, but they have not yet achieved the same degree of prominence in engineering applications.) Fuzzy logic helps solve the problem of human-machine communications (in querying experts) and formal symbolic reasoning (to a far less extent in current engineering applications).

In the past, when control engineers mainly emphasized the linear case and when AI was primarily Boolean, so-called intelligent control was mainly a matter of cutting and pasting: AI systems and control theory systems communicated with each other, in relatively ad hoc and distant ways, but the fit was not very good. Now, however, fuzzy logic and neuro control both build nonlinear systems, based on continuous variables bounded at 0 and 1. From the controller equations alone, it becomes more and more difficult to tell which system is a neural system and which is a fuzzy system; the distinction begins to become meaningless in terms of the mathematics. This moves us towards a new era, where control theory and AI will become far more compatible with each other. This allows arrangements like what is shown in Figure 2, where neuro control and fuzzy logic can be used as two complementary sets of tools for use on one common controller.

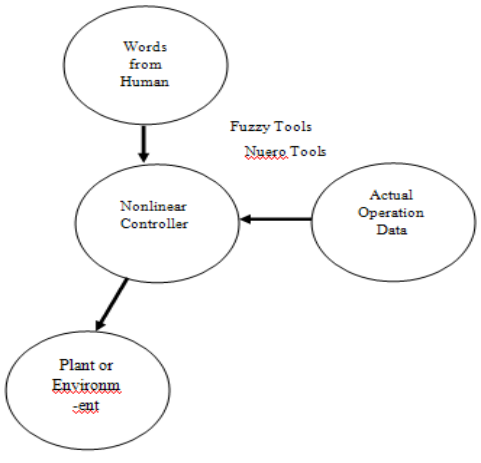


Figure 2: A way to combine fuzzy and neural tools.

## III. NEUROCONTROL: FIVE BASIC DESIGN APPROACHES

Neurocontrol is defined as the use of well-specified neural networks-artificial or natural-to emit actual control signals and are still based on five basic design approaches:

### *A. Supervised Control: Common Problems*

Supervised control seems very simple to implement, but in practice, it presents a number of challenges. The first challenge is to build up the database of sensor inputs and desired actions. If we already know what actions to take in a wide variety of situations, doesn't that mean that we already have an adequate controller? If so, what good is the neural net equivalent? In actuality, there are many situations where it is useful to copy a known controller with a neural network. For example, one might copy the skills of a human expert, so as to "clone" him or speed him up. Conventional expert systems copy what a person says to do, but supervised control copies what a person actually does (as recorded in the database). Likewise, supervised control can be used to copy a computer-based controller that runs very well on a Cray at slow speed but requires a smaller, faster copy for real-world applications.

For optimal performance, therefore, supervised control should not be treated as a simple exercise in static mapping. It should be treated as an exercise in system identification, an exercise in adapting a dynamic model of the human expert.

### *B. Common problems with Direct Inverse Control*

The challenge is to build a neural network that will input a desired 10 cation-X(t)-specified by a higher-level controller or a human, and output the control signals, u(t), which will move the robot arm to the desired location.

Most people using direct inverse control begin by building a database of actual X(t) and u(t) simply by flailing the arm about; they train a neural net to learn the inverse mapping from X(t) to u(t). In other words, they use supervised learning, with the actual X(t) used as the inputs and the actual u(t) used as the targets.

### *C. Adaptive Control and Neural Networks*

The ideal optimal controller should do as well as possible in coping with all three kinds of events; however, it is often good enough in practice just to handle type 1 or type 2 events, and there are severe limits to what is possible for any kind of control design in handling type 3 problems in the real world.

Real-time learning can help with all three types of events, but it is really crucial only for type 3. For type 1, it may be of minimal value. Type 3 events may also require special handling, using fast associative memories (e.g., your past experience flashes by in an instant as your plane dives) and idiosyncratic systems that take over in extreme circumstances. One way to improve the handling of type 3 events is to try to anticipate what kinds of things might go wrong, so as to make them more like type 2 events. The best way to handle type 1 events is to use a control system
which is explicitly designed to account for the presence of noise, such as an adaptive critic system.

### *D. Back propagating Utility and Recurrent Networks*

The back propagation of utility is an exact and straightforward method, essentially equivalent to the calculus of variations in classical control theory. The back propagation of utility can be used on two different kinds of tasks: (1) to adapt the parameters or weights, W, of a controller or Action network A (x, W); (2) to adapt a schedule of control actions over future time. The former approach-first proposed in 1974 [3]-was used by Jordan in his robot arm controller and by Widrow in his truck-backer-upper [4]. The latter approach was used by Kawato in his cascade method to control a robot arm [6] and by myself, in an official 1987 DOE model of the natural gas industry [5] involve optimization subject to constraints. This section will emphasize the former approach.

### *E. Adaptive Critics*

The adaptive critic family of designs is more complex than the other four. The simplest adaptive critic designs learn slowly on large problems but have generated many real-world success stories on difficult small problems. Complex adaptive critics may seem intimidating, at first, but they are the only design approach that shows serious promise of duplicating critical aspects of human intelligence: the ability to cope with large numbers of variables in parallel, in real-time, in a noisy nonlinear environment.

Adaptive critic designs may be defined as designs that attempt to approximate dynamic programming in the general case. Dynamic programming, in turn, is the only exact and efficient method for finding an optimal strategy of action over time in a noisy, nonlinear environment. The cost of running true dynamic programming is proportional (or worse) to the number of possible states in the plant or environment; that number, in turn, grows exponentially with the number of variables in the environment. Therefore, approximate methods are needed even with many small-scale problems.

Naturally there are many ways to combine the five basic designs in complex real-world applications. For example, there are many complex problems where it is difficult to find a good controller by adaptation alone, starting from random weights. In such problems, it is crucial to use a strategy called "shaping." In shaping, one first adapts a simpler controller to a simplified version of the problem, perhaps by using a simpler neurocontrol approach or even by talking to an expert; then, one uses the weights of the resulting controller as the initial values of the weights of a controller to solve the more complex problem. This approach can, of course, be repeated many times if necessary.

## IV. CONCLUSION

The term "intelligent control" has been used in a variety of ways, some very thoughtful, and some based on crude attempts to market aging software. To us, "intelligent control" should involve both intelligence and control theory. It should be based on a serious attempt to understand and replicate the phenomena that we have always called "intelligence" i.e., the generalized, flexible, and adaptive kind of capability that we see in the human brain.

Furthermore, it should be firmly rooted in control theory to the fullest extent possible; admittedly, our development of new designs must often be highly intuitive in the early stages, but, once these designs are specified, we should at least do our best to understand them and evaluate them in terms of the deepest possible mathematical theory.

## V. REFERENCES


[1] L. G. Kraft and D. Campagna, A summary comparison of CMAC neural network and traditional adaptive control systems. Neural Networks for Control, W. T. Miller, R. Sutton, and P. Werbos, MIT Press, Cambridge, MA, 1990.

[2] A. Guez and J. Selinsky, A trainable neuromorphic controller. Journal of Robotic Systems, August 1988.

[3] P. Werbos, Backpropagation through time: What it does and how to do it. Proceedings of the IEEE, October 1990.

[4] D. Nguyen and B. Widrow, The truck backer-upper: an example of self-learning in neural networks. In W. T. Miller, R. Sutton, and P. Werbos, op. cit. [1].

[5] P. Werbos, Maximizing long-term gas industry profits in two minutes in Lotus using neural network methods. IEEE Trans. Systems, Man, and Cybernetics, March/April 1989.

[6] M. Kawato, Computational schemes and neural network models for formation and control of multijoint arm trajectory. In W. T. Miller, R. Sutton, and P. Werbos, op. cit. [1].

[7] Commons, Grossberg, and Staddon, eds., Neural Network Models of Conditioning and Action, Erlbaum, Hillsdale, NJ, 1991.

[8] P. Werbos, Neurocontrol and fuzzy logic: Connections and designs. International Journal on Approximate Reasoning, January 1992.

[9] P. Werbos, Neurocontrol and related techniques. Handbook of Neural Computing Applications, Maren ed., Academic Press, New York, 1990.

[10] E. Sontag, Feedback Stabilization Using Two Hidden-Layer Nets, SYCON-90-11. Rutgers University Center for Systems and Control, New Brunswick, NJ, October 1990.

[11] T. Kohonen, The self-organizing map. Proceedings of the IEEE, September 1990.

[12] W. Y. Huang and R. P. Lippman, Neural net and traditional classifiers. In NIPS Proceedings 1987.

[13] N. DeClaris and M. Su, A novel class of neural networks with quadratic junctions. In 1991 IEEElSMC, IEEE Catalog No. 91CH3067-6, IEEE, New York, 1991.

[14] R. Jacobs et al., Adaptive mixtures of local experts. Neural Computation, 3:(1), 1991.