Position Paper: The Use of Expert Systems for Control System Design and Real-Time Control Systems [1]

James H. Taylor
Control Systems Laboratory
GE Corporate R & D
PO Box 8, Schenectady, NY 12301
email: taylorjh@ge-crd

First, my general conclusions related to both *control system design* and *real-time control systems* are that the use of artificial intelligence (AI), and expert systems (ES) in particular, offers a great deal with respect to the advancement of control technology **but** that this potential is not as great as has been suggested or promised by some workers in the field, even in principle. The problem in principle is that there seems to be a tendency to mythologize the potential capability of AI software and to undervalue the power of the human mind that the software is attempting to mimic. Simply stated, there is no magic in an ES - knowledge acquisition, coding, and validation is a major effort, always commensurate with the complexity of the task assigned to the ES. What AI technology has to offer here is a higher-level framework to carry out the development of a knowledge-based system (in comparison with hard-wired knowledge in a conventionally-programmed system). The benefits of a higher-level environment are quite widely understood to include a closer correspondence between the concept and the code, and therefore better understanding, maintenance, and extension of the software.

In practice, a major current impediment to bringing ES technology to bear on significant controls problems with a reasonable cost/benefit ratio is that the support in terms of ES shells or ES languages is still under development, and therefore somewhat rudimentary and unfriendly. There are many problems in terms of power, documentation, and support. Much work is being done with home-made or heavily customized ES shells; it is almost impossible to achieve a cost-effective solution in such an environment. In short, looking at both the theoretical and practical aspects of ES applications, my principal concern at this point in the development of "expert controls technology" is that a balanced, realistic perspective be conveyed to the controls community.

I will summarize specific points in each field below. In the panel discussion, I will present concrete examples to make these issues more concrete.

[1] Reformatted in LaTex, November 2014

1 Expert Systems for Control System Design

The idea of using an expert system as a higher-level user interface for control system design has become well established over the course of the last five years or so. The basic idea is that there is a lot of low-level nonsense that the user of existing Computer-Aided Control Engineering (CACE) software has to put up with that could be alleviated; common examples include:

1. reminding or helping a user deal with a complicated, rigid user interface, or replacing it;
2. executing or guiding the user in complicated but straight-forward higher-level procedures;
3. keeping track of alternative designs and design trade-offs;
4. catching and perhaps correcting software errors;
5. keeping track of a data base that may become quite large in a realistic control system design and validation project;
6. etc., etc.

An expert system that can deal meaningfully with most or all of these problems would be a substantial help to the user. It is important to keep the following points in mind, however, when one is concerned with developing an expert system that deals with these issues **realistically**:

1. In my experience, none of these functionalities are inherently deep in terms of the embedded knowledge. However, there is a great deal of work involved in them. In other words, the reasoning is shallow but the amount of detail is large.
2. The ES has to flexible enough to be supportive of the novice yet not patronizing or a nuisance to the more experienced user. This is a tall order!
3. One must beware of the "single scenario trap": An ES that can thread its way along a narrowly-defined path for a well-posed problem is a long way from being a useful system.

2 Expert Systems for Real-time Control

This application of AI is also pretty obvious, and substantial work (as well as a lot of talk!) has gone into this area in the past five years. The basic idea is that real-time

systems are often very complicated in terms of the logical/decision-making activity that must be built in to handle initializations, operating regime changes, exception-handling, failure detection, isolation, and accommodation, etc. This complex logic is often developed in a "seat of the pants" style which, in a conventionally-developed environment, can lead to a system that is cumbersome, messy, hard to understand, and very difficult to maintain or extend. This type of functionality can be implemented in an ES framework, with several advantages: a higher-level implementation, a better user interface, and less problems with respect to understanding, maintaining, extending, etc. ... assuming the ES is developed with reasonable care and discipline.

Despite this promise, much of what I have seen is quite unrealistic, both in terms of what is expected of this methodology (magic) and the cost (cheap). I maintain that both of these expectations are going to be troublesome in the short term, i.e., until expectations become more closely aligned with reality. The present problems I see are:

1. The "magic systems" syndrome - many of the expected benefits of real-time AI are not founded on a firm footing, like the existence of *real* intelligent solutions (the "Heck, that will be easy to do" assumption).
2. The "mimicking a human operator is straight-forward" syndrome, i.e., underestimating the capabilities of the person doing a seemingly routine task.
3. The "Cray-in-the-sky" syndrome, i.e., conceiving real-time AI solutions that cannot be obtained with realistic computational power.

In summary, I firmly believe that AI and controls technology can be combined to provide exciting new solutions to some very pressing problems. However, I feel that it is very important to emphasize that AI is **not** a panacea.