Fuzzy Logic Controllers: A Knowledge–Based System Perspective

Piero P. Bonissone

Artificial Intelligence Laboratory
GE Corporate Research and Development
Bldg. K1-5C32A, PO Box 8
Schenectady, NY 12301
Bonissone@crd.ge.com

Over the last few years we have seen an increasing number of applications of Fuzzy Logic Controllers. These applications range from the development of auto-focus cameras, to the control of subway trains, cranes, automobile subsystems (automatic transmissions), domestic appliances, and various consumer electronic products.

A Fuzzy Logic Controller is a knowledge based system in which the knowledge of process operators or product engineers has been used to synthesize a closed loop controller for the process. We will compare the development and deployment of Fuzzy Logic Controllers (FLC) with that of Knowledge Based System (KBS) applications.

Traditional controllers are derived from a mathematical model of the open-loop process to be controlled, following classical control theory techniques. FLCs are typically derived from a knowledge acquisition process (or are automatically synthesized from a self-organizing control architecture). In either case, the result of the synthesis is a Knowledge Base (KB), rather than an algorithm. The KB consists of a set of fuzzy-rules (rules and termsets), which is evaluated by an interpreter. The interpreter is composed of a quantification (or fuzzification) stage, an inference engine (or fuzzy matcher), and a defuzzification stage.

We will analyze FLCs according to three organizing layers typically used in describing Knowledge Based Systems: knowledge representation, inference, and control. In the knowledge representation layer we will describe fuzzy state vectors, term-set of linguistic values, and fuzzy production rules. In the inference layer we will provide a geometric interpretation (for the disjunctive case) of the generalized modus ponens, and describe the inference process based on fuzzy predicate evaluation, rule Left Hand Side (LHS) evaluation, rule detachment, and rules aggregation. In the control layer we will show three different defuzzification methods and we illustrate meta-reasoning capabilities (supervisory mode).

FLC interpreters are used during the development phase of a FLC application to provide inference traceability (transparency), which facilitates the KB design, implementation, and refinement. However, the use of an interpreter requires the evaluation of all the rules in the KB at every iteration.

Therefore, after a functional validation (stability or robustness analysis), the KB is compiled like a programming language or a traditional knowledge base application, and a simpler run-time engine is used for deployment. The result of this compilation process is a look-up table that allows for a faster, more efficient execution that can be performed by simpler processors. Not only is the response time reduced, but the memory requirements are so drastically decreased that it is possible to implement the FLC using very small amounts of memory. This feature enables us to build inexpensive FLCs for cost-sensitive applications.

In summary, we consider a Fuzzy Logic Controller to be a high level language with its local semantics, interpreter, and compiler, which enables us to quickly synthesize non-linear controllers for dynamic systems.