Kosko B.: Fuzzy Engineering, Prentice Hall, 1997

Rule patches lead to the fundamental fuzzy approximation theorem [14–16] of fuzzy engineering. A fuzzy system F approximates a function f by covering the graph of f with rule patches and averaging patches that overlap. The next chapter explores this result in detail and many other chapters apply it to real problems. The approximation is uniform and that allows the user in theory to pick the approximation error level in advance. The search for such rules may not be so easy in practice and makes up one of the main research areas of fuzzy engineering. The approximation theorem has a constructive proof that suggests that data clusters can define rule patches if there are enough data and if the data reflect the unknown system or process f.

The patch covering of fuzzy systems also leads to their greatest weakness: exponential rule explosion [14]. The number of rules a fuzzy system F needs to cover the graph of a function f grows exponentially with the number of input and output dimensions n and p. More input variables can give a better causal model of a process but at a higher cost in knowledge acquisition and computation. Suppose n input variables each have m fuzzy subsets on their axes. Then it takes m^n rules just to cover the input space. Optimal rules can make the best of a fixed rule budget but may be hard to find. Lone optimal rule patches cover the turning points or extrema of f but that knowledge may be of no help if the user does not know at least the rough shape of the approximand f. The goal of fuzzy learning theory is to shape and move the rule patches to optimal locations. But the neural learning schemes discussed in Section 1.5 often involve their own exponential computational complexity and require a stream of accurate data that the user may not have.

1.4 FUZZY SYSTEMS IN COMMERCIAL PRODUCTS

Fuzzy products use fuzzy systems in their microprocessors. Some use a fuzzy chip but most just reprogram their existing chip. The fuzzy inference scheme discussed previously and in Section 1.6 takes up only a few lines of software code. Japan leads the world in fuzzy products and holds over a thousand patents in Japan on fuzzy designs. Most applications have had few inputs and outputs and this has helped keep the rule explosion manageable. Fuzzy systems are also highly nonlinear and model-free. So it often takes extensive computer simulations or real tests to study their sensitivity and stability in control applications.

The most famous fuzzy application is the fuzzy control of the subway system in the Japanese city of Sendai. The subway runs on a 13.6-kilometer route with 16 stations. Hitachi [35] programmed rules in a fuzzy system to brake the subway and a fuzzy system to speed and slow it: If the train speed exceeds the limit speed then slow the speed. If the train is in the allowed zone then brake slightly (rather than accelerate). The fuzzy system gives a smoother ride than did human control and it outperforms standard PID (proportional-integrative-differential) controllers in smoothness of braking and acceleration and in electric power consumption. It also stops the subway with greater accuracy.

The Japanese government required extensive testing of the fuzzy system. In the 1980s Hitachi Corporation ran over 300,000 simulations of the fuzzy system and ran over 3,000 runs on real but empty subways before the government let the fuzzy system replace human controllers in 1987. Today the fuzzy system runs the subway during peak hours. Humans still control the subway in nonpeak hours to keep up their operating skills.

Several firms in Japan and Korea manufacture fuzzy washing machines. A fuzzy washing machine gives a finer wash than does a "dumb" washing machine with fixed commands. It tailors the wash to each set of clothes and changes the wash strategy as the clothes clean. That helps prevent cloth damage and under- or overwashing. A fuzzy washing machine turns sensor data into wash times and wash cycles. A pulsing optical sensor measures the murk or clarity of the wash water and measures how long it takes a stain to dissolve or saturate in the wash water. Mud and dirt break down quickly. Oil stains break down more slowly. Some machines use a load sensor to change the agitation rate or water temperature. Some shoot bubbles into the wash to help dissolve dirt and detergent.

The simplest systems map water clarity and saturation time to wash time. They may use as few as 10 fuzzy rules: If the water clarity is low and the saturation time is short then wash long. If the water clarity is low and the saturation time is long then wash very long. If the water clarity is high and the saturation time is short then wash short. Most wash water is clear to some degree and saturates quickly or slowly to some degree. So each second the machine's microprocessor fires most of the rules to some degree. Engineers run thousands of experiments in advance to tune the fuzzy sets *short*, *medium*, *high*, *long*, and others to new shapes. Some systems use neural or statistical schemes to tune the fuzzy-set curves.

Fuzzy cameras and camcorders map image data to lens settings. The first fuzzy camera was the Canon handheld H800 model in 1990. It tuned the autofocus with 13 fuzzy rules. Sensors measure the image clarity and the change in the image clarity in six parts of the image. The 13 rules take up only 1.1 kilobytes of memory and convert the 12 types of sensor data to new lens settings. The first fuzzy camcorder was the Sanyo-Fisher 8mm FVC–880 model. It tuned the autofocus with only 9 fuzzy rules. The rules tune the lens setting with the relative contrast between six regions of the image. The center region counts most and the border regions count least. Nonfuzzy camcorders just weigh and add the image intensity in the six regions.

The fuzzy rules adjust the camcorder lens with the relative contrast between the six regions. Some regions are much brighter or a little less brighter than others or are almost equal in brightness. Matsushita (Panasonic) adds more rules to cancel the image jitter that a shaking hand causes in small camcorders. The fuzzy rules tell movement within the frame from movement of the whole frame: If all the image points move in the same direction then the hand shakes. The system compares the current frame with past frames to look for global movement. Math models of image jitter tend to model and cancel only a few types of jitter.

A few fuzzy rules map sensor data to control settings in many household products and car systems. A fuzzy dryer converts load size and fabric type and the flow of hot air to drying times and drying strategies. A fuzzy vacuum sweeper uses changes in dust flow to judge if the floor is bare or carpeted. A pulsing infrared light emitting diode measures the dust flow. A simple 4-bit microprocessor converts the dust flow to suction power and beater-bar settings. Mitsubishi and Korea's Samsung report energy savings of 40% to 100% with their fuzzy vacuum sweepers over their nonfuzzy designs.

Hitachi, Sanyo, Sharp, and Toshiba have designed fuzzy microwave ovens. A fuzzy microwave oven measures infrared light patterns, temperature, humidity, and change in food shape. Fuzzy rules associate these conditions with whether the food is frozen or thawed and with how well cooked it is. This in turn maps to power and cooking times and control actions like hot-air blowing or roasting.

Nissan has patented a fuzzy antiskid braking system, fuzzy transmission system, and fuzzy fuel injector. The antiskid system tries to brake right up to the point of locking when the car slows down too fast. The fuel injector adjusts the fuel flow with fuzzy rules in an on-board microprocessor. Sensors measure the throttle setting, manifold pressure, water temperature, and car rpm. A second set of fuzzy rules times the ignition with sensor data on the car rpm, water temperature, and oxygen concentration. Mitsubishi has developed an omnibus fuzzy system that controls a car's suspension, transmission, steering, traction, and air conditioner.

Michio Sugeno of the Tokyo Institute of Technology has designed one of the most complex fuzzy systems. It uses fuzzy voice commands like "up" and "land" and "hover" to control an unmanned helicopter [28]. Pilots must train for weeks to hover in a helicopter. The first version was a scaled helicopter of length 3.58 meters and that weighed 20 kilograms and had a 1-meter rotor. The new version has a 3-meter rotor. The system takes 13 inputs and gives four control outputs in terms of the elevator, aileron, throttle, and rudder. One set of fuzzy rules controls the higher-level task of navigation: If the flight mode is hover and if the flight state is forward then the standard trim is such and such mix of pitch angle, roll angle, elevator offset, and aileron offset. A second set of rules controls the lower-level task of stabilization: If the body pitches then control the elevator in reverse. An Omron 16-bit microprocessor houses the entire fuzzy system. Cavalcante [3] has designed a like fuzzy system to control a simulated helicopter.

Fuzzy systems can also manage information systems. Omron uses a fuzzy system to manage five medical databases in a health management system for large firms. The fuzzy system uses 500 rules to give each of up to 10,000 patients a health diagnosis and a personal health plan to help prevent disease and to help each patient stay fit and reduce stress. Hitachi uses 150 or so fuzzy rules to trade in Japanese bonds and bond futures. Yamaichi Securities uses hundreds of rules to manage a stock fund. The rules model private firms and the Japanese economy and large industries such as textiles and automobiles and consumer electronics.

In most fuzzy products and systems an expert has given rules to the engineer. A few fuzzy rules give a quick approximation of the control system. Then the hard task is to tune the rules and fuzzy sets. That can take days or weeks of software trial and error. There are now many software packages that help users tune fuzzy systems and port them to microprocessors. Often there are no experts to ask for rules. Then the user must turn to statistical or neural schemes to learn the rules from training data.