Truncation Depth Rule-of-Thumb for Convolutional Codes

The new rule is more accurate and tight at high signal-to-noise ratios.

NASA's Jet Propulsion Laboratory, Pasadena, California

In this innovation, it is shown that a commonly used rule of thumb (that the truncation depth of a convolutional code should be five times the memory length, m, of the code) is accurate only for rate 1/2 codes. In fact, the truncation depth should be 2.5 m/(1 – r), where r is the code rate. The accuracy of this new rule is demonstrated by tabulating the distance properties of a large set of known codes. This new rule was derived by bounding the losses due to truncation as a function of the code rate.

The bound derives from a result on random trellis codes in G.D. Forney, Jr.'s "Convolutional codes II: Maximum likelihood decoding," Information and Control, vol. 25:222-266 (1974). An (M, ν) trellis is a trellis corresponding to a shift register of length ν where each register contains a M-vector and the input is an M-ary sequence (the corresponding trellis contains M^ν states). An (M, ν, n) trellis code augments an (M, ν) trellis by assigning n channel symbols to each edge. The rate of the code is r = log₂(M)/n bits/symbol. A random trellis code is an (M, ν, n) trellis in which each channel symbol on each edge is chosen randomly and independently according to some distribution p. When M = q^k, the (M, ν, n) trellis corresponds to a rate log₂(q)/k nonsystematic convolutional code over GF(q) with k equal constraint lengths ν = v, 1 ≤ i ≤ k. The memory of this code is m = max_ν=v. It is presumed that the code is decoded via the Viterbi algorithm with decisions on edges of the trellis made after a delay of T trellis stages. A truncation error occurs when an incorrect edge is chosen that would not have been chosen with an infinite truncation depth.

In the case of punctured codes, the truncation depth on the mother code trellis should be increased as the rate increases. Punctured code can be created by forming (q^k, ν, n) code by puncturing a (q^k, ν, n) mother code, where k divides k and ν = ν/k. This resulting code is the daughter code.

Efficient Method for Optimizing Placement of Sensors

This systematic method supplants ad hoc placement and exhaustive-search optimization methods.

NASA's Jet Propulsion Laboratory, Pasadena, California

A computationally efficient method has been developed to enable optimization of the placement of sensors for the purpose of diagnosis of a complex engineering system (e.g., an aircraft or spacecraft). The method can be used both in (1) designing a sensor system in which the number and positions of sensors are initially not known and must be determined and (2) adding sensors to a pre-existing system to increase the diagnostic capability.

The optimal-sensor-placement problem can be summarized as involving the following concepts, issues, and subproblems:

- Degree of Diagnosability — This is a concept for characterizing the set of faults that can be discriminated by use of a given set of sensors.
- Minimal Sensor Set — The idea is one of finding a minimal set of sensors that guarantees a specific degree of diagnosability.
- Minimal-Cost Sensors — In a case in which different sensors are assigned with different costs, a minimal-cost sensor set is one that results in the minimum cost.

A System of Four Multiplier and Three Adder Gates serves as an example for illustrating the concept of ARRs and a signature matrix. In this example, there are three sensors that measure the variables f, g, and h. Each element of the matrix is 1 or 0 if the ARR listed in the row containing that element is or is not, respectively, affected by a fault in the gate listed in the column containing that element.