NASA TECH BRIEFS

Algorithm for Rapid Acquisition of a PPM Optical Signal

Data frames can be synchronized with part of the synchronization sequence.

NASA's Jet Propulsion Laboratory, Pasadena, California

An algorithm for rapid acquisition of a pulse-position-modulation (PPM) optical data signal implements a pattern-matching technique for synchronization of receiver timing with the temporal boundaries of data frames. Synchronization is necessary because in PPM, information is conveyed by the time slot during which a pulse is detected. Fast acquisition of a signal depends on detection of pulses in noise, and on correct estimation of the times of detected pulses. To facilitate synchronization at the receiver, a transmitter periodically inserts a prescribed sequence of pulses — the synchronization sequence or word — into the transmitted data stream. Older PPM-signal-acquisition algorithms are based on correlations and depend on reception of the full synchronization word (128 bytes long in some applications). The present algorithm is more computationally efficient and is capable of achieving synchronization with part of the synchronization word — typically with as few as 2 to 6 bytes.

The algorithm involves, among other things, selective elimination of unlikely starting points for a sequence of received pulses detected in the presence of noise. It is assumed that the pulse-detection threshold of the receiver is set at a level that corresponds to a specified bit-error probability. The receiver is switched on at a random time, and a search for the synchronization sequence is started. The objective of the algorithm is to uniquely identify the temporal location of a received sub-sequence of the synchronization sequence.

Initially, the algorithm seeks all possible candidate sub-sequences that match a received 2-byte sequence. The process continues with the reception of additional bytes; the algorithm seeks matches with 3-byte sub-sequences, 4-byte sub-sequences, and so forth, until it eliminates all candidate sub-sequences that do not match and arrives at a single unique sub-sequence and thereby establishes the receiver timing relative to the entire synchronization sequence. The figure presents an example of this process in the case of a 32-byte synchronization sequence.

When real data are interleaved between periodic transmissions of the synchronization sequence, data-bit patterns can sometimes be identical to sub-sequences of the synchronization sequence. Random bit flips caused by noise can also cause bit patterns other than the desired ones to alias as synchronization sub-sequences. To reduce the incidence of such errors, one can require confirmation in the form of additional received bytes beyond those needed to establish a unique synchronization sub-sequence; the additional bytes are said to constitute a confirmation sequence. The probability of false synchronization is inversely proportional to the length of the confirmation sequence.

The procedure for confirmation immediately follows the identification of a unique sub-sequence and determination of its starting time. The unique sub-sequence enables the algorithm to predict the arrival of the next pulse in the known synchronization sequence. If the next detected pulse arrives at the expected time, confirmation continues with the next pulse, and so on, for as many pulses as required for the confirmation sequence. If all predictions and observations have been

found to match when the confirmation procedure has been completed, then synchronization is deemed to have been achieved and the temporal boundaries of data frames are calculated. If a mismatched pulse is detected during the confirmation procedure, then the algorithm resets.

WS: Word Synchronization

PPL: Possible Pattern Location

Received pattern: **1 0 1**   (Two Bytes Received)
WS : 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1
PPL: 0 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0
            ↑   ↑ ↑   ↑

Received pattern: **1 0 1 1** (Three Bytes Received)
WS : 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1
PPL: 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0
            ↑   ↑

Received pattern: 1 0 1 1 0 1 (Four Bytes Received)
WS : 1 0 0 0 0 1 0 1 1 0 1 0 1 0 0 0 1 1 1 0 1 1 1 1 1 0 0 1 0 0 1 1
PPL: 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
            ↑   ***Frame Synchronization Complete***

Received Byte Patterns are compared with sub-sequences of a 32-bit synchronization sequence. Matches with sub-sequences of increasing length are sought until the received pattern matches a unique sub-sequence.