
▶ Protograph-Based Raptor-Like Codes

The proposed codes have the advantage of low-complexity encoder and decoder implementation.

NASA's Jet Propulsion Laboratory, Pasadena, California

Theoretical analysis has long indicated that feedback improves the error exponent but not the capacity of point-to-point memoryless channels. The analytic and empirical results indicate that at short blocklength regime, practical rate-compatible punctured convolutional (RCPC) codes achieve low latency with the use of noiseless feedback. In 3GPP, standard rate-compatible turbo codes (RCPT) did not outperform the convolutional codes in the short blocklength regime. The reason is the convolutional codes for low number of states can be decoded optimally using Viterbi decoder. Despite excellent performance of convolutional codes at very short blocklengths, the strength of convolutional codes does not scale with the blocklength for a fixed number of states in its trellis.

Protograph-based (PB) Raptor-like codes can provide good performance in an incremental redundancy scheme with noiseless feedback over an additive white Gaussian noise channel. Additionally, these codes are also desirable

for other applications where there is a need for simple generation of various code rates.

The proposed codes are based on protograph construction and they represent a novel contribution. In the original Raptor code, the redundant bit generation is based on a random selection of precoded bits that are produced from an unstructured LDPC (Low Density Parity Check) code. These redundant bits are selected based on some optimized distribution. Due to the nature of random selection, the original Raptor code required some additional information to be transmitted to the receiver in order to enable the decoding process. In the proposed codes, the redundant bits are generated based to optimized protograph structure with degree-1 nodes. Thus they do not need any additional information to be transmitted to the receiver. The proposed codes with protograph-based structure have the advantage of low-complexity encoder and decoder implementation. The proposed codes were designed for short block sizes, but a similar construction

method can be applied to longer block lengths for other applications.

Hybrid ARQ (hybrid automatic repeat request — HARQ) is an error control method. In standard ARQ error detection, symbols such as cyclic redundancy check (CRC) are added to the information data. In HARQ, forward error correction code such as LDPC code symbols are also added to the existing error detection symbols, such that small random errors are corrected without retransmission, and major errors are corrected via a request for retransmission. The hybrid scheme performs better than standard ARQ in poor signal conditions. The proposed protograph-based Raptor-like codes can be used with HARQ.

This work was done by Dariush Divsalar of Caltech; and Tsung-Yi Chen, Jiadong Wang, and Richard D. Wesel of UCLA for NASA's Jet Propulsion Laboratory. For more information, contact iaoffice@jpl.nasa.gov.

The software used in this innovation is available for commercial licensing. Please contact Dan Broderick at Daniel.F.Broderick@jpl.nasa.gov. Refer to NPO-48128.

▶ Fuzzy Neuron: Method and Hardware Realization

Simple and effective learning functions and adaptive elements can be placed into small hardware systems to include instruments for space, bioimplantable devices, and stochastic observers.

John H. Glenn Research Center, Cleveland, Ohio

This innovation represents a method by which single-to-multi-input, single-to-many-output system transfer functions can be estimated from input/output data sets. This innovation can be run in the background while a system is operating under other means (e.g., through human operator effort), or may be utilized offline using data sets created from observations of the estimated system. It utilizes a set of fuzzy membership functions spanning the input space for each input variable. Linear combiners associated with combinations of input membership functions are used to create the output(s) of the estimator. Coefficients are adjusted online through the use of learning algorithms.

This innovation has been demonstrated to be capable of creating usable

models that can effect any number of complex transfer functions such as a continuous exclusive OR function, time domain (slew rate) filter, automatic gain controller, non-linear algebraic function calculator, and more. This innovation was created specifically for embedment within microcontrollers, allowing for simple and effective placement of learning functions and adaptive elements into small hardware systems to include instruments for space, bioimplantable devices, stochastic observers, etc.

Small spaceflight (and other) instruments have been confined to simple systems utilizing microcontrollers and a microcontroller core. Since most learning algorithms typically reside in larger computational frames and are rather complex (neural nets, for example), a

simpler solution to self-learning, auto-adaptive systems would be attractive for smaller embodiments. Fuzzy logic systems lend themselves well to microcontrollers, but adaptive fuzzy systems also require a good deal of computational power. Thus, the simpler components of both fuzzy systems (input membership functions) and the back error propagation neural net (the linear combiner) were selected and fused into a simple two-layer system that can be easily embedded into common microcontrollers.

The training method used is an LMS (least mean square) algorithm based on a modification to the Widrow-Hoff learning algorithm. Coefficients and constants for each linear combiner were initialized to random values. Training data from observations of a user's

input(s) to a system and the resultant output(s) in real time or a posteriori, or from software-generated data sets, were presented to the system, which generated outputs. Once a system is learned, the coefficients and constants can be

frozen and the algorithm embedded in an application.

This work was done by Michael J. Krasowski and Norman F. Prokop of Glenn Research Center. Further information is contained in a TSP (see page 1).

Inquiries concerning rights for the commercial use of this invention should be addressed to NASA Glenn Research Center, Innovative Partnerships Office, Attn: Steven Fedor, Mail Stop 4-8, 21000 Brookpark Road, Cleveland, Ohio 44135. Refer to LEW-18887-1.

➤ Kalman Filter Input Processor for Boresight Calibration

The new software brings this technology to the industrial level.

NASA's Jet Propulsion Laboratory, Pasadena, California

Ka-band ranging provides the phase center (PC) to phase center range, which needs to be converted to the center of mass (CM) to center of mass range. Nominally, both PC and CM lie on the line connecting the spacecraft GRAIL A and GRAIL B. In this case, the conversion should be done simply by adding the CM-to-PC distance L to the measured range for both spacecraft. However, due to various technical reasons, such as displacement of the true CM from its nominal position in the SRF, or spacecraft attitude fluctuations, the PC and CM define a unit vector that may be different from the nominal line of sight. The objectives of the software are to determine the actual line of sight direction for each spacecraft and correct the previously recorded range data, and to provide instructions for how to ma-

neuver each spacecraft to make necessary attitude corrections.

While elements of this approach have been used for the boresight calibration in the GRACE project, the new software brings this technology to the industrial level. It is now fully documented and can be used by people other than its developers. This innovation provides graphic outputs and log files that are critical for quick analysis and troubleshooting. In addition to the line of sight direction, the software allows one to evaluate the CM-PC base length, which is important when the PM location is subject to variations (e.g., due to fuel depletion).

This software is implemented in Python and offers excellent cross-platform porting possibilities. It is very versatile, and may be applied under various circumstances and for other related pur-

poses. This innovation is capable of combining the input data from several calibration maneuvers, evaluating individual range biases, and compressing the time stamps. It uses Lagrange interpolation for the orbit data, and a unique quaternion-interpolating algorithm for interpolating the attitude data. As a result, data files with different data rates and independent time stamps can be handled together.

This work was done by Dmitry V. Strelkov, Gerhard L. Kruizinga, Meegyeong Paik, Dah-Ning Yuan, and Sami W. Asmar of Caltech for NASA's Jet Propulsion Laboratory. For more information, please contact Brian Morrison at Brian.A.Morrison@jpl.nasa.gov.

This software is available for commercial licensing. Please contact Dan Broderick at Daniel.F.Broderick@jpl.nasa.gov. Refer to NPO-48479.

➤ Organizing Compression of Hyperspectral Imagery to Allow Efficient Parallel Decompression

Higher compression factors can be attained.

NASA's Jet Propulsion Laboratory, Pasadena, California

A family of schemes has been devised for organizing the output of an algorithm for predictive data compression of hyperspectral imagery so as to allow efficient parallelization in both the compressor and decompressor. In these schemes, the compressor performs a number of iterations, during each of which a portion of the data is compressed via parallel threads operating on independent portions of the data. The general idea is that for each iteration it is predetermined how much compressed data will be produced from each thread.

A simple version of this technique is applicable when the image is divided into "pieces" that are compressed inde-

pendently. As an example, for a compressor that does not make use of inter-band correlation, a piece could be defined to be an individual spectral band, or a fixed number of bands. In the technique, the compressed output for a piece is comprised of multiple "chunks." The concatenated chunks for a given piece form the compressed output for the piece. Most of the compressed image is produced in multiple iterations, where during a given iteration, one chunk is produced for each piece. Prior to the start of an iteration, chunk sizes are calculated for each piece. The chunks can be produced or decompressed in parallel. It is noted that it is not specified how

much of the image data will go into a chunk, and in fact a chunk may contain incomplete portions of encoded samples (at the chunk's start or end). The compressor iterates the process of deciding on chunk sizes and producing chunks for each piece of the requested size, until compression of each piece is almost finished. At that point, the remainder of the pieces is compressed serially without a target chunk size.

Typically, the chunk size calculation should seek to balance the progress through each piece, i.e., to leave equal numbers of samples remaining in each piece; a suggested procedure has this aim. A key requirement on the chunk