Project no: FP6-2002-IST-C 020023-2

Project title: FlexCode

Instrument: STREP

Thematic Priority: Information Society Technologies

D2.1 Generic Binary Input Soft Output (BISO) Channel Model

Due date of deliverable: 2007-07-01
Actual submission date: 2007-MM-DD

Start date of project: 2006-07-01 Duration: 36 Months

Organization name of lead contractor for this deliverable: RWTH Aachen University

Revision X.X

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

D2.1 Generic Binary Input Soft Output (BISO) Channel Model

The Channel Model

For the design of digital communication systems extensive simulations have to be performed before the hardware implementation. In order to simulate the real-world channel conditions and to guarantee reproducible results well-defined abstracted channel models of manageable complexity are needed. In WP2.1 a flexible and widely usable channel model is developed to examine the behavior of the FlexCode transmission scheme in various channel situations. The FlexCode channel model is a binary input soft output (BISO) channel model, describing the equivalent channel from the binary input of the modulator to the soft output of the demodulator. The soft values are required for the advanced source and channel (de-)coding algorithms which are developed in FlexCode. Both the base channel and the modulation scheme can be adapted and controlled individually by a set of parameters that determines the characteristics of the BISO channel model. During the first project year a complete BISO channel model has been developed and implemented as indicated in Figure 1. As the objective of the FlexCode project is to design and implement a flexible coding scheme with instantaneous adaptation to varying channel conditions, a forward and backward side channel for header and control information is provided to emulate packet erasures, whereas the payload bits are transmitted over the BISO channel model. Transmission of header and control bits over the BISO channel with dedicated error protection is out of the scope of FlexCode. The approach indicated in Figure 1 allows to consider either traces of measured packet losses or to implement stochastic Markov modeling of packet erasures.

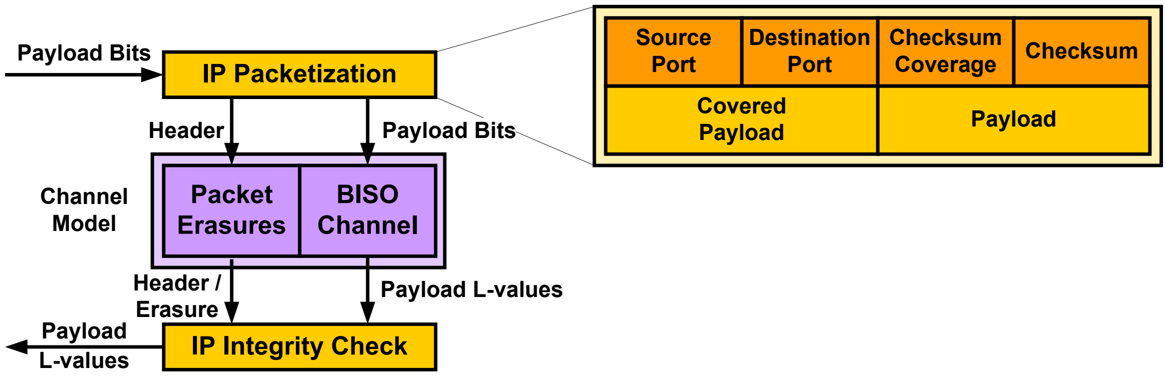

Figure 1: IP extension of the generic BISO channel model with UDP Lite packet structure

The kernel of the generic (M,N)-BISO channel model, which is depicted in Figure 2, consists of a Discrete Memoryless Channel (DMC) that simulates the transitions of the M modulation symbols to the N demodulation symbols. N may be larger than M in order to provide quantized soft information (reliability information) at the demodulator output. The demodulation symbols may thus be from a larger set than the modulation symbols. In the extreme case, the demodulation symbols are elements of the continuous complex space. The higher the resolution of the demodulation symbols the more information can be drawn from them and can be exploited in iterative source and channel decoding algorithms or for channel estimation. The transition matrix which statistically describes the transitions between modulation and demodulation symbols is parameterized such that it can be adjusted to any measurements of real channels.

In addition, a BISO channel model with unquantized channel output has been implemented, providing AWGN, Rayleigh fading, and Rice fading channels.

The characteristics of channels with memory can be achieved by appropriate dynamic control of the transition matrix of the (M,N)-BISO channel model and the fading factor of the unquantized model.

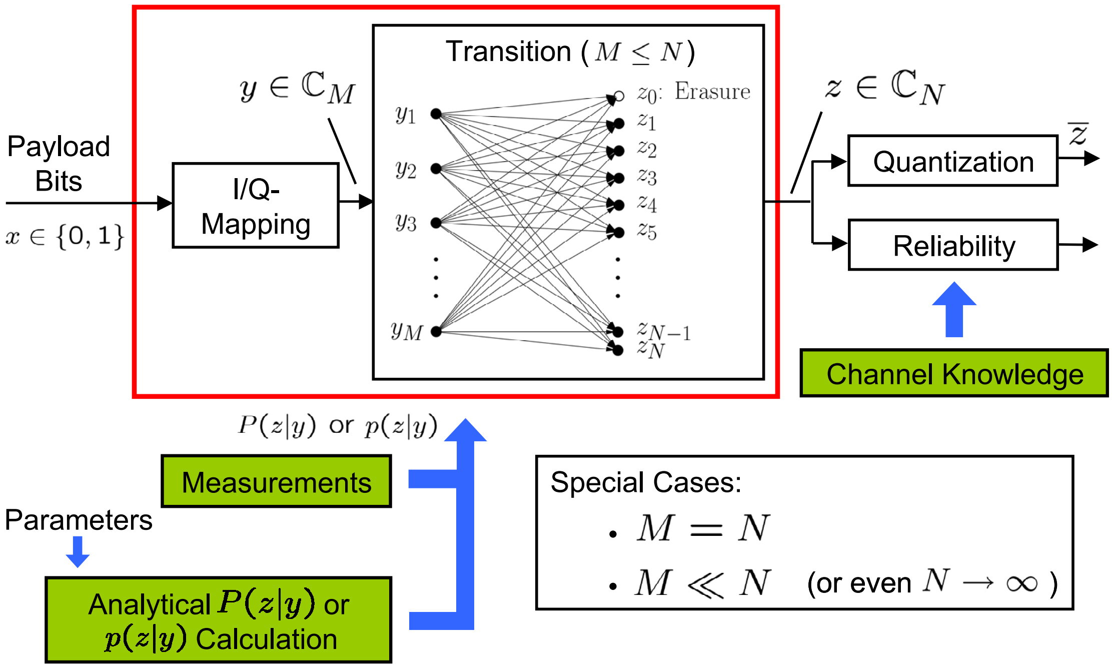

Figure 2: The generic FlexCode BISO channel model scheme

Supplemental to circuit switched transmission also packet switched transmission is provided by integrated IP modules that are implemented before and after the BISO channel. On the transmitter side packets are generated of the channel coded data and a header is prepended which contains relevant control bits, such as the source and destination port or a checksum. Packet erasures are realized by a separate erasure module as depicted in Figure 1.

There exist different strategies on how to deal with disturbed packets. One promising approach is UDP Lite [1] (Lightweight User Datagram Protocol – a variation of UDP), which allows to protect only parts of the payload by a checksum. On the receiver side an integrity check is performed so that disturbed packets can be separated out. Even if a packet contains errors it is not discarded if the errors occur in the part of the payload not covered by the checksum. Since the fixed or mobile access to the internet infrastructure may allow access to soft information, the demodulator can deliver soft values which may be exploited in source and channel decoding. The IP channel model is therefore designed to handle both, hard and soft values. However, so far the integrity check is based on hard values, i.e., a hard decision is made for the packet whereof the checksum is computed. If the checksum is not valid, the packet is discarded, if it is valid, soft values are transmitted to the channel decoder. In order to generalize the integrity check towards soft value computation, which is commonly used in demodulation, channel decoding and source decoding, we have derived the mathematical prerequisites for a soft integrity check. One may say that in general soft bit computation exceeds the performance of hard bit computation, as hard bits usually contain far less information than the corresponding soft values. The benefits of soft integrity check will be evaluated by theoretical considerations and intensive simulations. To have a common soft bit structure, the complete generic BISO channel model has been made compatible to the ITU-T G.192 [2] Error Insertion Device (EID) which is depicted in Figure 3.

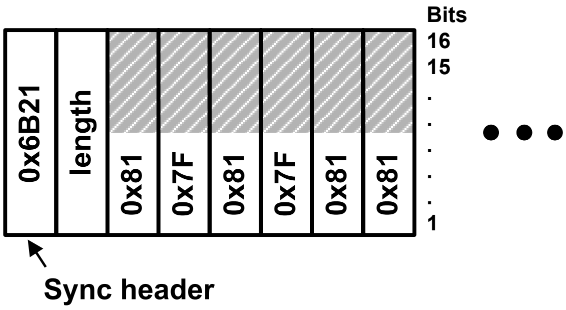

Figure 3: Soft bit structure of the ITU-T G.192 Error Insertion Device

The soft bits given by the ITU-T G.192 structure can be mapped to probabilities or L-values which are required in iterative source-channel decoding. The mapping between these quantities can be according to Table 1. The resolution of the L-values is very high and therefore no relevant quantization loss occurs. This enables a close to maximum exploitation of soft information in the following iterative receiver.

hard bit	soft bit in 2`s complement	soft bit decimal	$P(x=+1)$	$L(x) = \ln\left(\frac{p(x=+1)}{p(x=-1)}\right)$
0 $\longrightarrow$ +1	0111 1111	+127	1.0	$+\infty$
	...	...	...	...
	0000 1111	+15	0.559	0.2373
	...	...	...	...
	0000 0001	+1	0.504	0.0157
	0000 0000	0	0.500	0.0
	1111 1111	-1	0.496	-0.0157
	...	...	...	...
	1111 0001	-15	0.441	-0.2373
	...	...	...	...
0 $\longrightarrow$ -1	1000 0001	-127	0.0	$-\infty$

Table 1: Soft bit format and corresponding probabilities and L-values

Channel State Estimation

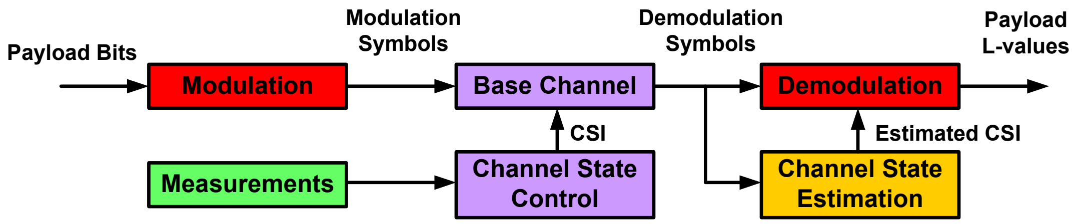

Figure 4: Scheme of the BISO channel model with channel state estimation

Since the goal of FlexCode is to flexibly adapt both, the transmitter and the receiver to a time-varying channel, it is required to provide channel state information to the various stages of the receiver (demodulator, channel decoder, soft decision source decoder). However, in real communication systems the channel is unknown to the transmitter and the receiver unless channel state estimation techniques are utilized. Many modern source and channel (de-)coding algorithms require a quite exact knowledge of the current channel state information (CSI) for a good performance, it is very important to have accurate estimates of the CSI. As a first step various channel estimation techniques [3], [4] differing in accuracy and computational complexity have been implemented that estimate the signal-to-noise ratio (SNR) of AWGN channels for the BPSK, QPSK and 8PSK modulation schemes.

Most utilized algorithms are maximum likelihood estimators approaching the Cramer-Rao lower bound (CRLB) very closely with increasing SNR. Other estimators extract the information on the SNR from the second and fourth order moment of the signal. The latter algorithms have a very low complexity, but they also only perform well in high SNR regions respectively in low SNR regions for long data frames. All yet considered algorithms are non-data-aided, i.e., they estimate the SNR from the unknown received symbols without transmission of pilot symbols. However, we will also study algorithms for data-aided channel state estimation, so that potentially available pilots are exploited in order to increase the estimation accuracy.

As an example, we consider the following transmission model, where $z_k = ay_k + n_k$, resp. $\mathbf{z} = a\mathbf{y} + \mathbf{n}$ is the received signal, with the transmitted symbol vector $\mathbf{y} = [y_1, \ldots y_N]^T$, $y_k \in \{-1, +1\}$, a positive channel gain a, the noise vector $\mathbf{n} = [n_1, \ldots n_N]^T$ with additive white gaussian noise samples n_k, and the channel state information $\boldsymbol{\theta} = [a, \sigma^2]^T$ which is estimated in form of the SNR $\rho = \frac{a^2}{\sigma^2}$.

Some of the implemented estimation algorithms, e.g., for AWGN and BPSK are the following:

- Maximum likelihood method 'MLD' (D for „deterministic", [5], [6])
- Method of moments 'MM' (e.g., [7])
- Maximum likelihood method 'MLR' with expectation maximization (R for „random", [3])

whose performances can be compared by means of Figures 5 – 7. The performance of estimators can be determined for different aspects. Figure 5 shows, that these MLD and MLR estimation algorithms are biased, and thus do not deliver the correct estimate for low SNRs even for an infinite number of data, while the MM algorithm is unbiased and delivers the correct estimate. However, considering Figure 6 it becomes clear, that the MM algorithm has a much higher variance than the MLR algorithm. Figure 7 shows the convergence speed of the three algorithms. As already expected from Figure 5 the MLD estimator does not converge to the correct SNR, while the other two algorithms approach the true value after several frames. The question which estimator is finally utilized is to answer application specifically, since besides the different estimation accuracies, they also have different computational complexities. For estimating high SNRs the MLD algorithm works sufficiently well, while for low SNRs the considered algorithm would rather be MM or MLR.

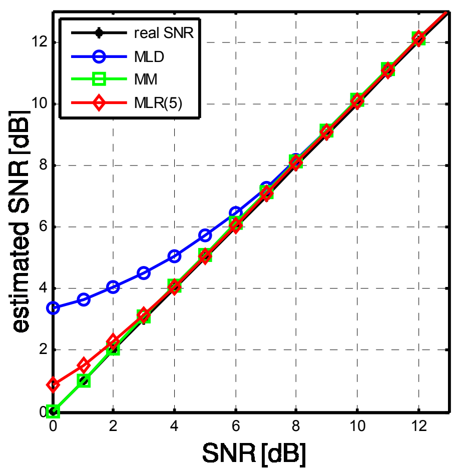

Figure 5: Performance of CSI estimators

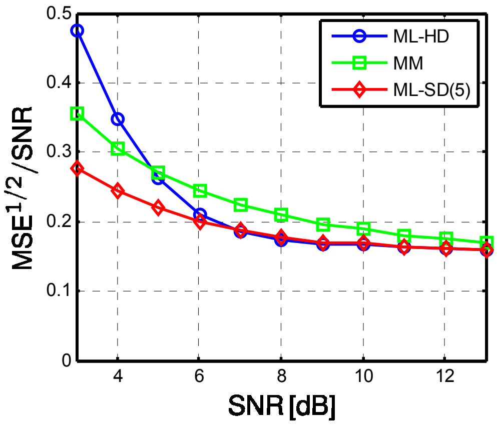

Figure 6: Normalized root mean square error of CSI estimators

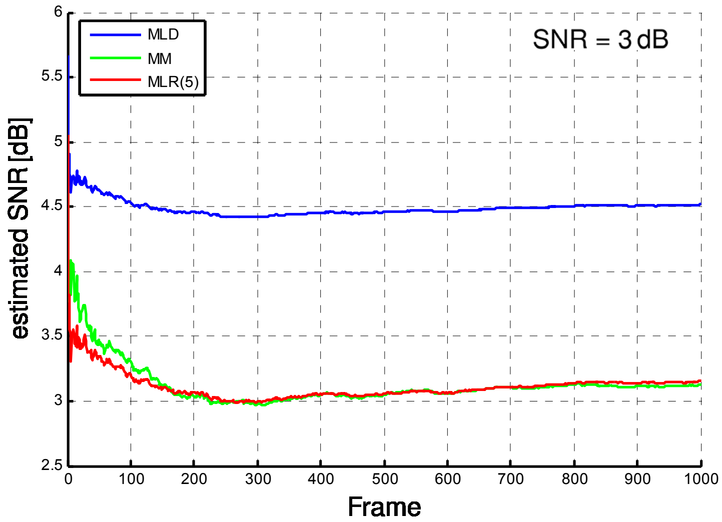

Figure 7: Convergence of CSI estimators

Software package: Generic Channel Model

According to the project plan, a first release of the generic channel model is available to the FlexCode project partners by the end of June. The software package has been written in C++. This deliverable comprises a first version of the channel model as illustrated in Figure 1. The model consists of two separate kernels for the transmission of header and payload bits. The first release offers the following features:

- Memoryless fading AWGN channels
- Phase modulation schemes (BPSK, QPSK, 8PSK)
- Connection oriented transmission with hard decision, soft decision and erasures on bit level
- Packet oriented transmission with packet erasures and variable delays
- Manual settings of typical channel scenarios (BISO transition probabilities)
- Static control of the model by channel traces provided by measurements

During the following project year additional modulation schemes shall follow (e.g., 16QAM), as well as various optimized modulation mappings. Furthermore, the model will be extended towards channels with memory. The extended model will be driven dynamically by corresponding random generators, which are fitted to real channel measurements.

The input and output format is compatible with the format of the ITU-T G.192 Error Insertion Device and may be considered as a future extension of the corresponding ITU standard.

Furthermore, a first version of a separate software package for channel state estimation has been implemented based on the described estimation algorithms. The channel state estimator is capable of tracing varying AWGN-Fading conditions. While the performance of the different estimators has been evaluated under static channel conditions, the investigation of the estimation accuracy under dynamic channel conditions is still ongoing.

Both, the channel model and the channel estimator are controlled by configuration files. The specification and implementation of a graphical user interface has not yet been completed and will be part of the next deliverable.

[1] L.-A. Larzon, M. Degermark, S. Pink, L.-E. Jonsson, G. Fairhurst, "The Lightweight User Datagram Protocol (UDP-Lite)", *RFC 3828*, July 2004.

[2] ITU-T. Recommendation G.192, "A Common Digital Parallel Interface for Speech Standardization Activities", ITU-T, Geneva, November 1995.

[3] A. Wiesel, J. Goldberg, H. Messer, "Non-Data-Aided Signal-to-Noise-Ratio Estimation," *Proceedings of International Conference on Communications (ICC)*, New York, NY, USA, 2002

[4] G. Ren, Y. Chang, H. Zhang, "A New SNR's Estimator for QPSK Modulations in an AWGN Channel," IEEE Transactions on Analogue and Digital Signal Processing, Volume 52, Issue 6, June 2005

[5] R. B. Kerr, "On Signal and Noise Level Estimation in a Coherent PCM Channel," IEEE Transactions on Aerospace and Electronic Systems, vol. 2, pp. 450-454, July 1966

[6] R. M. Gagliardi and C. M. Thomas, "PCM Data Reliability Monitoring through Estimation of Signal-to-Noise-Ratio," IEEE Transactions on Communications, vol. 16, pp. 479-486, June 1968

[7] D. R. Pauluzzi and N. C. Beaulieu, "A Comparison of SNR Estimation Techniques for the AWGN Channel," IEEE Transactions on Communications, vol. 48, no. 10, pp. 1681-1691, Oct. 2000