Mercury Computer Systems, Inc.

Matlab README

FM3TR Waveform Reference Implementation

SDR Forum Contract

March 23, 2007

Revision 1.0

Table of Contents

1 Introduction

This document describes the Matlab simulation of the Reed-Solomon library as used in the data mode of the FM3TR waveform.

2 Description of Files

2.1 fm3tr_monte_carlo.m

This file contains the script for the fm3tr data waveform BER testing. This is the main script for establishing the Monte Carlo simulations, and includes the parameters for execution.

2.2 fm3tr_code.m

This file contains the matlab function fm3tr_code(BER,numBlockMessage).

The function implements the full encode/decode chain of the Fm3tr channel coding, with introducing errors. By changing the profileCode variable in the code to 1, Matlab profiling can also be turned on.

The processing chain is as follows:

Block Enc. → 7to5 Bit Conv. → Hop Enc. → Add Errors (Channel)
Hop Dec. → 5to7 Bit Conv. → Block Dec. → Check Results

2.3 rs_encode.m

This file contains the matlab funcion rs_encode(T,n,k,b) where:

T : the message to be encoded, and it needs to be a Galois field (GF) element created by the gf() method of the Matlab Communication toolbox.
n : the number of symbols in the encoded codeword
k : the number of symbols in the information codeword
b : defined by the formula in the BBC paper

Unlike the Matlab library function rsenc, this function can calculate the RS codes for both odd and even values of (n-k), the number of redundant symbols in the encoded codeword.

The encoding is done as explained in the BBC White Paper.

The function uses the rs_init function in order the initialize the lookup tables that are saved as global variables.

2.4 rs_decode.m

This file contains the matlab function [T_synth failures]= rs_decode(R,n,k,b) where:

R : needs to be a GF object to be decoded.

n : the number of symbols in the encoded codeword. This number can be less than 2m-1 in case of a shortened RS code.

k : the number of symbols in the decoded message.

b : the factor in (X-ab)*(X-a (b+1))*...*(X-a(b+n-k-1)), defined in the generator polynomial, where b is an integer ('a' is read as 'alpha'). If it is not provided, b is assumed to be 1 by default

The function implements Reed Solomon decoding using the Euclidean Algorithm. It works for both when (n-k) is odd and even.

This function uses rs_init function in order the initialize the lookup tables that are saved as global variables and rs_syndromes function to calculate the syndrome values of the given received polynomial.

2.5 rs_init.m

This file contains the function Rs_init(N,K,m,p,b).

This function initializes the lookup tables that the other functions use, such as:

- RS Generator Polynomial, based on n, k, b, and the primitive polynomial of the information codeword
- alpha-i lookup table
- log alpha(i) lookup table

After initializing those tables, rs_init saves them in a global variable, which is of struct type.

2.6 rs_syndromes.m

This file contains the function Rs_syndromes(R,n,k,b) which calculates the syndrome values using the Horner's method. This function uses the rs_init function in order the initialize the lookup tables that are saved as global variables.

2.7 rs_errors.m

This file contains the function [R E]= rs_errors(T,err,ErrorType) where the ErrorType determines how the errors will be calculated. If it is 'bler', then the err variable is treated as the block error rate that will be used to generate errors. If it is 'num', then the err variable represents a fixed number of errors (but at random locations) that will be introduced to codeword T.

3 How to use

The file 'fm3tr_monte_carlo.m' contains two lines which change the scope of the simulation:

```
BER = logspace(-2, -6, 18)';
N_Trials = 2;
```

The first establishes the range and resolution over which the input bit error rate (BER) is chosen. The second defines the number of trials over which the simulator averages the errors.

By default, 18 BER points are chosen logarithmically between 10^-6 and 10^-2, the results of which are averaged over 2 independent trials.

Once the simulator has completed, two plots are presented:

1. Probability of decoding a block error vs. input BER
2. Probability of decoding a block error vs. input BLER

where 'BLER' refers to the block error rate. The relation between BER and BLER can be described with the following formula:

BLER = 1 - (1-BER)^16

4 Matlab Dependencies

The included Matlab files and functions rely on the Communications Toolbox for Galois field mathematics, necessary for the Reed-Solomon encoders and decoders.