Radio Data Signal Transmission With Software Defined Radios

Lab Report

Daniel
Simon Schmitt
Technische Universtität Darmstadt
your.name@stud.tu-darmstadt.de
simon_johannes.schmitt@stud.tu-darmstadt.de

ABSTRACT

Answer the following questions with roughly one sentence each:

One-page summary:

- What is the topic of your main seminar paper?
- What problem does it solve?
- Why is that topic/problem important?
- What methodologies do the authors apply?
- What are the main contributions of the paper?
- What are the key findings/results of the paper?

Final seminar paper:

- What is your research question?
- Why are that question and your topic important?
- How did you proceed to answer the question?
- What (do you think) is the answer to your question?
- Give an overall opinion on your topic.
- If you have results, describe them.
- What is the impact of the answers to your questions?

1 INTRODUCTION

Radio signals have been used for more than a century now, to broadcast information wirelessly. The best known one is FM radio, as it can be received by everyone with a common radio receiver. While FM radio was originally developt to broadcast music and news reports from different radio stations, more enhanced receivers allowed to process additional information (e.g traffic updates in car radios) next to the existing ones. Therefore, the *Radio Data System (RDS)* was invented. In this lab report we will give an introduction of RDS, how it can be implemented and how future work might look like.

2 IMPLEMENTATION

In this Section we describe how RDS signals can be generated using MATLAB.

2.0.1 Preparation: Baseband Coding. Before the actual implementation we have to define and format the information we are about to generate.

Setup.

- The programme service name should be "#SEEMOO#"
- We want to transmit a Mono signal without Artificial Head
- The signal is not compressed
- We use a static PTY which is set to Pop Music
- We want to transmit Music
- We do not carry traffic announcements
- Our radio station is in Germany
- We cover a local area
- You can choose your own programme reference number

SEEMOO Seminars Winter 2017/18, February 2018, Darmstadt, Germany

Using the standard documentation of the RDS, we can create the following table, which represents the given information in a baseband coding.

Group 1 Block 1	1 1 0 1	0 0 0 0	0 0 0 0	0 0 0 1
Group 1 Block 2	0 0 0 0	1 0 0 1	0 1 0 0	1 0 0 0
Group 1 Block 1	1 1 0 1	0 0 0 0	0 0 0 0	0 0 0 1
Group 1 Block 4	0 0 1 0	0 0 1 1	0 1 0 1	0 0 1 1
Group 2 Block 1	1 1 0 1	0 0 0 0	0 0 0 0	0 0 0 1
Group 2 Block 2	0 0 0 0	1 0 0 1	0 1 0 0	1 0 0 1
Group 2 Block 3	1 1 0 1	0 0 0 0	0 0 0 0	0 0 0 1
Group 2 Block 4	0 1 0 0	0 1 0 1	0 1 0 0	0 1 0 1
Group 3 Block 1	1 1 0 1	0 0 0 0	0 0 0 0	0 0 0 1
Group 3 Block 2	0 0 0 0	1 0 0 1	0 1 0 0	1 0 1 0
Group 3 Block 3	1 1 0 1	0 0 0 0	0 0 0 0	0 0 0 1
Group 3 Block 4	0 1 0 0	1 1 0 1	0 1 0 0	1 1 1 1
Group 4 Block 1	1 1 0 1	0 0 0 0	0 0 0 0	0 0 0 1
Group 4 Block 2	0 0 0 0	1 0 0 1	0 1 0 0	1 0 1 1
Group 4 Block 3	1 1 0 1	0 0 0 0	0 0 0 0	0 0 0 1
Group 4 Block 4	0 1 0 0	1 1 1 1	0 0 1 0	0 0 1 1

2.0.2 Task1: Bitstream Generation. Within this first task we initialize the message we wanna send. Therefore, we import the bit stream we previously prepared and append the corresponding checksum and offset. After decoding the message in the first test we get the following ouput, which shows the hexadecimal, binary and decoded representation.

Decoder Output.

```
d001_0de 094b_3e2 d001_372 4f23_212
1101_0000_0000_0001__00_1101_1110
0000_1001_0100_1011__11_1110_0010
1101_0000_0000_0001__11_0111_0010
0100_1111_0010_0011__10_0001_0010
00B (BASIC) - PI:D001 - PTY:
Pop Music (country:DE/LY/YU/__/__, area:Local, program:1)
==>#SEEMOO#<== -  -  -Music-STEREO - AF:p
```

2.0.3 Task2: Differential Encoding. Since we are about to use frequency modulation (FM) in one of the next steps it is recommended to perform a differential encoding. The problem with modulation is that a receiver of a signal can not determine the logic assigned to a phase shift, as it might be introduced by the wireless channel.

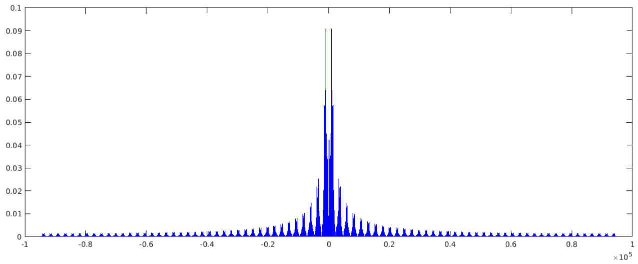

Figure 1

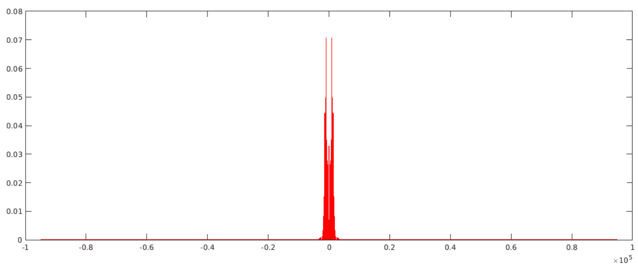

Figure 2

Differential encoding provides unambigiuous signal reception, because the encoding of the data depends not only on the current signal state (bit), but also on the previous one. ($y_i = y_{i-1} \bigoplus x$).

2.0.4 Task3: Manchester Encoding. *Manchester Encoding*, also called **Phase Encoding (PE)**, is a line code. The information is represented by the edge of a signal that corresponds to the clock (generated in the first two lines of Task3). A binary zero becomes a low to high transition, a binary one a high to low transition. This type of representation is named after G.E. Thomas. After the differential encoded bitstream has been mapped to Manchester Encoded symbols, we get a RDS signal. However, as shown in Figure1 this signal also contains frequencies we actualy dont need. Therefore, we apply a lowpass filter and normalize the resulting signal.

```
bit_clk = [ones(80,1);-ones(80,1)];
bit_clk = repmat(bit_clk,length(bitstream_differentially_encoded),1);
rds = kron(bitstream_differentially_encoded, [ones(80,1);-ones(80,1)]) ...
    + kron(~bitstream_differentially_encoded, [-ones(80,1);ones(80,1)]);
f6 = fir1(300,2400/fs*2);
rds_filt = filter(f6,1,rds);
rds_filt = rds_filt/max(rds_filt);
```

2.0.5 Task4: Generate Baseband Signal. We will now generate the baseband signal. This signal represents the actual data we wanna transmit. First of all, we have to change the frequency of the current signal to 57 kHz, since this is the expected frequency for a RDS signal. Therefore, we multiply the RDS signal with a 57 kHz sine wave. To generate a complete baseband signal we also have to insert a pilot and an audio signal. The pilot signal is genarated with a 19 kHz sine wave and it is used to upconvert the RDS signal. The audio signal is genarated as a mono signal, which means that we sum up the left and the right audio channel at a frequency of 0.3 to 15 kHz. The spectrum of the final baseband signal is shown in Figure3.

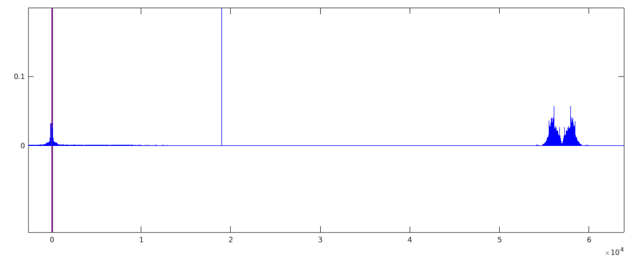

Figure 3

Figure 4

2.0.6 Task5: Frequency Modulation.

3 FUTURE WORK

REFERENCES