Get Start with gr-cdma

Yu Wang
umwangyu@umich.edu

July 23, 2016

Contents

Chapter 1

Introduction

1.1 What is gr-cdma

Project gr-cdma is the physical layer of CDMA communication. This project is developed based on GNURadio[1]. gr-cdma is created and maintained by Prof. Anastasopoulos. The whole source codes are on github `https://github.com/anastas/gr-cdma`. This project is to build a communication block which employs DS-CDMA technique to build the link. The user does not have to know the details of the schemes, but could transmit reliably through this physical layer protocol.

1.2 What is GNURadio

GNURadio[1] is a open-source *unix based platform which enables developers and researchers design and test their self-built communication or signal processing system. The whole platform is written by c++. Developer could use c++ or python to customize their own block. gr-cdma is a customized block as well.

1.3 What is the Current State of gr-cdma

Till 17-Jun-2016, the main part of this project is finished. The next step is to optimize the working strategy for several block to make the system more robust in field test. This document is an introduction of what this project is, how to get start with GNURadio and gr-cdma and what are the issues to be solved.

Chapter 2

Working Strategy of gr-cdma

In this chapter, we will discuss the working strategy of gr-cdma, the principle of each main block. Mathematic models and analyses are involved to describe the situation.

2.1 The diagram of The Tx & Rx

Here are the digrams for Tx and Rx.

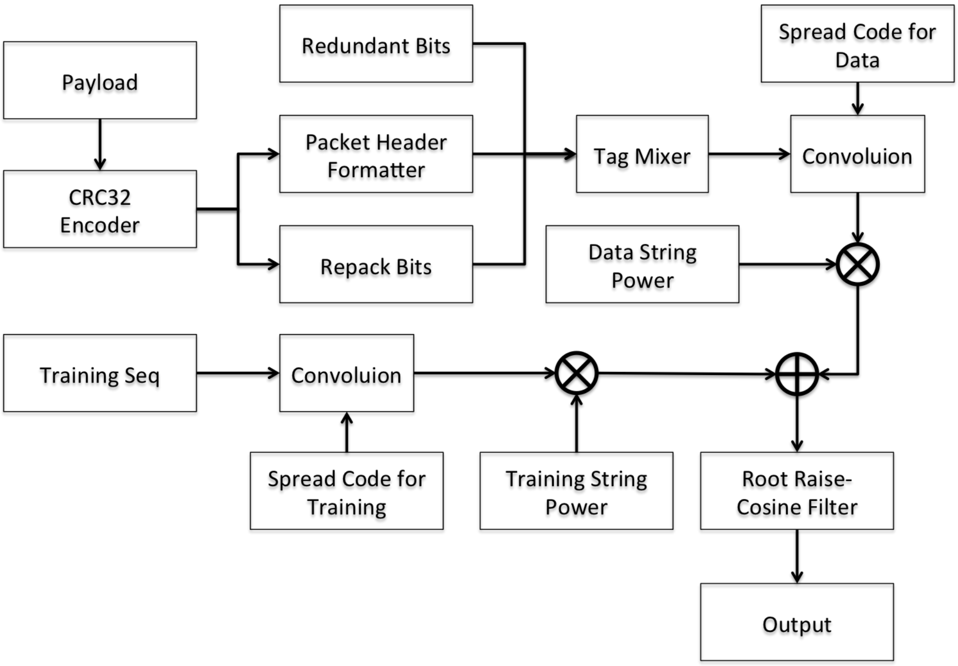


Figure 2.1: Diagram for Tx

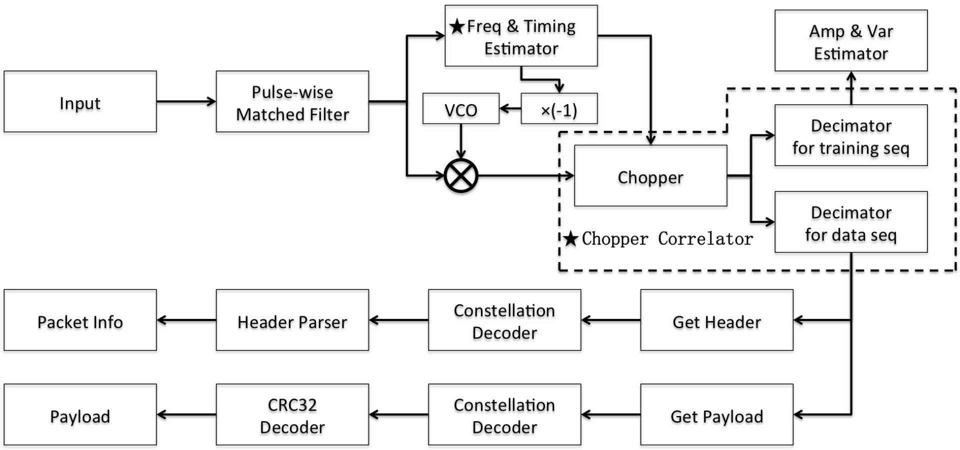


Figure 2.2: Diagram for Rx

From fig 2.1 and fig 2.2, you may find that the system is a little bit more complicated than you thought. And also, the receiver is bigger than the transmitter. In fact, it is for the reason that the receiver will deal with no only the decoding part, but also the synchronization problem. which is actually the most difficult part for this system.

2.2 Transmitter

The duty for transmitter is to package the payload or the input data, build the packet structure, generate the training sequence, modify the power for each section and form the waveform.

2.2.1 Spreading Code

Similar to other CDMA system, one of the essential part of the system is about the spreading code. The quality of the spreading code will affect greatly about the performance of synchronization parts. Intuitively, we would like to use codes like m-sequences and gold sequence, which has high auto-correlation and flat cross-correlation value.

The most famous spreading code is the m-sequence.

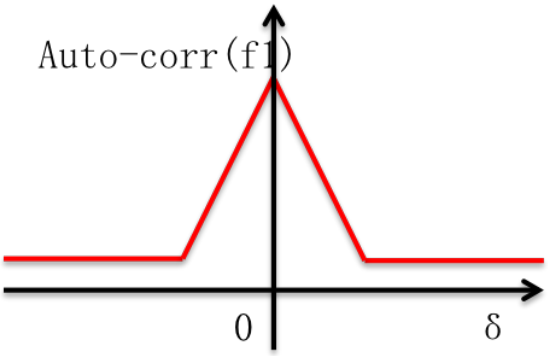
2.2.1.1 M-Sequence

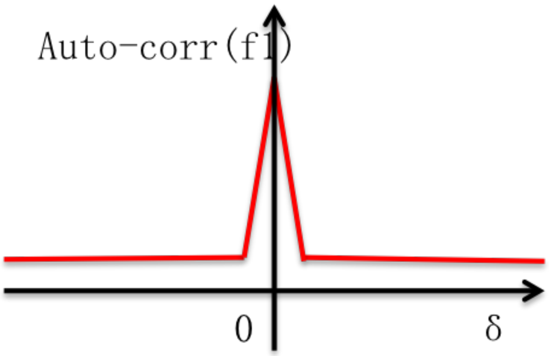
M-sequence is a kind of sequnece with very good property. for m-sequnece $p(k)$,

$$p(k) * p(k+\delta_k) = \begin{cases} m & \delta_k = 0 \\ -1 & \delta_k \neq 0 \end{cases} \tag{2.1}$$

m is the length of sequence, δ_k ranges from $-m+1$ to $m-1$.

Here is the table for some m-sequence, table 2.1





2.2.2 Gold Sequence

But also as shown in the table above, the number of m-sequence for certain length is not that much. So, some other sequences like “Gold sequences” are generated. “Gold sequence” is actually the shift-add of two m-sequence. When the shift space for tow m-sequence is different, more relatively good sequences are generated.

2.2.2.1 Spreading Seuquence used in gr-cdma

But for gr-cdma, we adopt sequence of length 8. From the diagram, we know that a functioning system needs two spreading sequences for “Training” and “Data”. So we only need two relatively good sequences. Because the length of codes are not long enough, so brutal search is feasible. For current system, the two sequences are as shown in table 2.2

2.2.3 Packet Structure

Packeting is also one main part for Transmitter. In physical layer, transmitting purely meaningless 0s and 1s reliably and efficiently is good. But for a system, we may care the index of each sequence, which is good for placing different pieces in order, modulation scheme, adaptive modulations may be used, and so on. The training and data packet structure for gr-cdma is like figure 2.4 and figure 2.3:

Table 2.1: M-sequence feedback connections generated using LFSR generator

Number of shift registers, L	Code length $N_c = 2^l - 1$	Feedback Taps for M-sequence	No. of M-sequence codes
2	3	[2,1]	1
3	7	[3,2]	2
4	15	[4,1]	2
5	31	[5,3], [5,4,3,2], [5,4,2,1]	6
6	63	[6,1], [6,5,2,1], [6,5,3,2]	6
7	127	[7,1], [7,3],[7,3,2,1], [7,4,3,2], [7,6,4,2], [7,6,3,1], [7,6,5,2], [7,6,5,4,2,1], [7,5,4,3,2,1]	18
8	255	[8,4,3,2], [8,6,5,3], [8,6,5,2], [8,5,3,1], [8,6,5,1], [8,7,6,1], [8,7,6,5,2,1], [8,6,4,3,2,1]	16
9	511	[9,4], [9,6,4,3], [9,8,5,4], [9,8,4,1], [9,5,3,2], [9,8,6,5], [9,8,7,2], [9,6,5,4,2,1], [9,7,5,3,1], [9,8,7,6,5,3]	48
10	1023	[10,3], [10,8,3,2], [10,4,3,1], [10,8,5,1], [10,8,5,4], [10,9,4,1], [10,8,4,3], [10,5,3,2], [10,5,2,1], [10,9,4,2], [10,6,5,3,2,1], [10,9,8,6,3,2], [10,9,8,7,6,5,4,3], [10,8,7,6,5,4,3,1]	176

Table 2.2: Spread Codes for gr-cdma

Type of Sequence	Sequence
Training Sequence	[+1, +1, +1, +1, -1, +1, +1, -1]
Data Sequence	[-1, +1, -1, +1, -1, -1, -1, -1]

In current version, because the number of binary bits for each Segment of the frame is set through “python/cdma_parameter.py”, the length for these part is

2.2.4 Power Control

As shown in the figure 2.1, training sequence and data sequence are two separate branches. In reality, the power a transmitter could provide is certain, so before adding them up, the power arrangement should be considered. In “python/cdma_parameter.py”, variable “training_percent” is to control the percentage of power used for Training sequence.

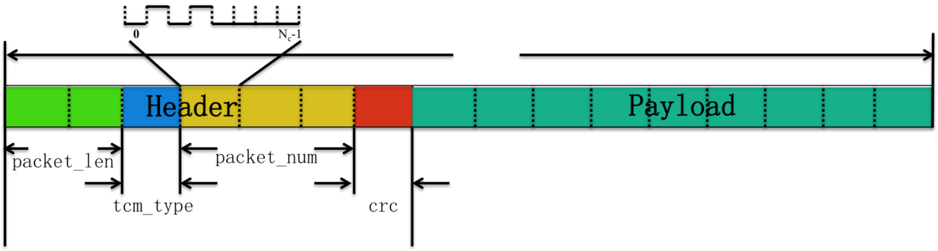


Figure 2.3: Data Frame

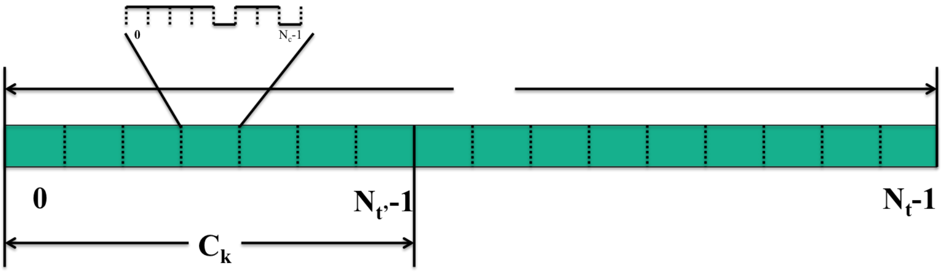


Figure 2.4: Training Frame

Table 2.3: Length for Each Part of Frame

Frame Type	Segment Type	Segment Length
Training	/	260
Data	packet length	12
Data	tcm type	4
Data	packet index	16
Data	header crc	8

Bibliography

[1] GNURadio. Gnuradio, 2016.