

DMT: IMPLEMENTATION^{*}

Chris Sramek

This work is produced by OpenStax-CN X and licensed under the
Creative Commons Attribution License 1.0[†]

Abstract

Describes our implementation of a DMT system.

1 DMT System

Our DMT system can be represented by the following block diagram:

^{*}Version 1.3: Dec 18, 2003 10:51 am +0000

[†]<http://creativecommons.org/licenses/by/1.0>

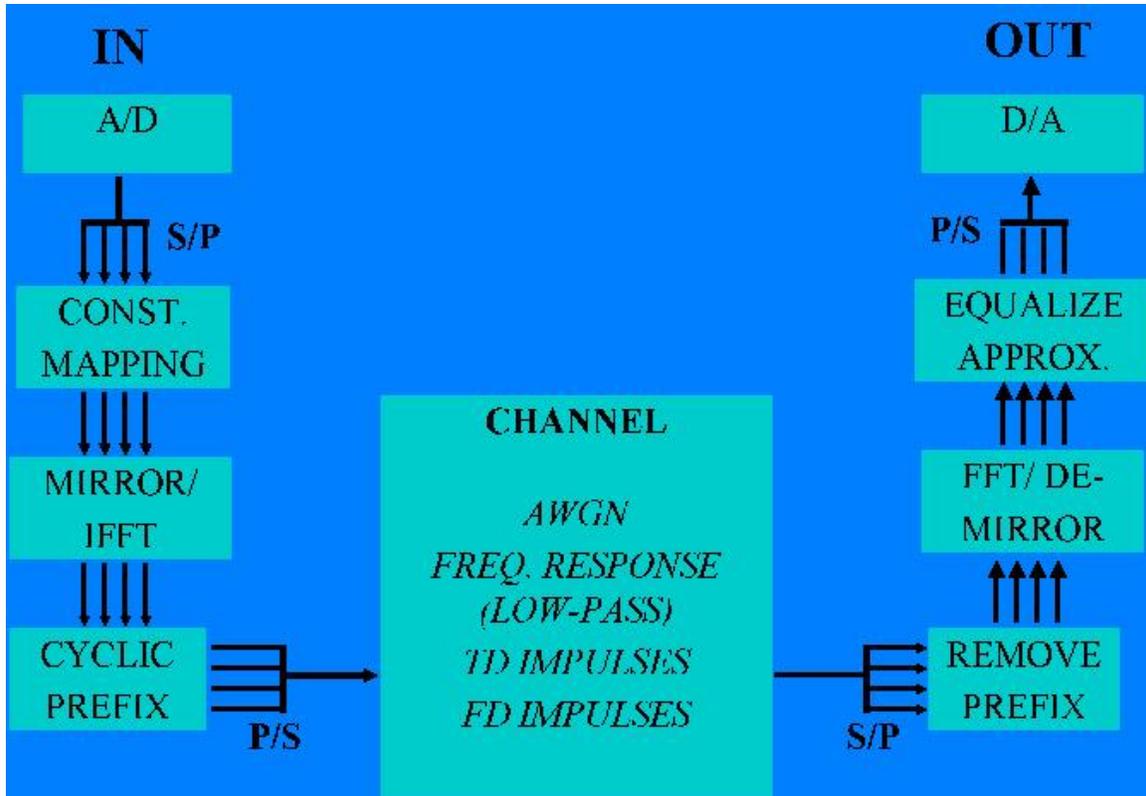
 Block Diagram


Figure 1

 Visit the following links to find out more about the particular subsystems:

- A/D & D/A¹
- Serial/Parallel, Parallel/Serial²
- Constellation Mapping³
- Mirror/IFFT, De-Mirror/FFT⁴
- Cyclic Prefix⁵
- The Channel⁶
- Equalization & Approximation⁷

¹<http://cnx.org/content/m11722/latest>
²<http://cnx.org/content/m11723/latest>
³<http://cnx.org/content/m11724/latest>
⁴<http://cnx.org/content/m11725/latest>
⁵<http://cnx.org/content/m11762/latest>
⁶<http://cnx.org/content/m11727/latest>
⁷<http://cnx.org/content/m11728/latest>

2 Operation of the System

Before we talk about our implementation, we need to describe how DMT may be used to send and receive an analog signal:

- A sampled analog signal passes through an A/D converter⁸ (Q bits/sample) where it is represented as a bitstream.
- The bitstream is divided into S length N parallel blocks with a serial-to-parallel converter⁹.
- Each block undergoes a constellation mapping¹⁰, where segments of bits (length B) are represented as single complex values in a constellation.
- The mapped blocks are mirrored¹¹ to give them conjugate symmetry (which increases their lengths to $2N/B$), then the IFFT of each block is taken.
- A cyclic prefix¹² (length C) is inserted at the beginning of each block to combat problems introduced by the channel.
- The blocks (now known as **symbols**) are concatenated¹³ to form a length $S(2N/B+C)$ time domain signal, which is transmitted into the channel¹⁴.
- On the other side of the channel¹⁵, the received signal is again broken up¹⁶ into S parallel blocks.
- The cyclic prefixes¹⁷ are removed, the FFT of each block is taken and each is de-mirrored¹⁸ so that the block length is once again N/B .
- Because the channel filtered the time domain signal and added noise, each block is equalized¹⁹ through multiplication by the inverse transfer function of the channel (which is known by the receiver).
- The complex numbers in each block, which have been distorted by noise, are approximated²⁰ by values in the original constellation. A de-constellation mapping then occurs, which converts the complex values back to bits and increases the block length back to N .
- The blocks of bits are concatenated²¹ back into a single bitstream, which then undergoes a D/A conversion²² back to a sampled analog signal. This is an approximation of the original signal.

3 Implementation

We implemented our DMT system through a collection of MATLAB functions. Some of them, such as the IFFT and FFT, were standard built-in functions while others, such as the constellation mapping and approximation routines, were coded from scratch. We directed the use of each function with a separate script, `run.m`, which allowed us to easily change input parameters. The MATLAB code for each function can be found below:

MATLAB code used:

- `run.m`²³
- `constbuilder.m`²⁴

⁸<http://cnx.org/content/m11722/latest>

⁹<http://cnx.org/content/m11723/latest>

¹⁰<http://cnx.org/content/m11724/latest>

¹¹<http://cnx.org/content/m11725/latest>

¹²<http://cnx.org/content/m11762/latest>

¹³<http://cnx.org/content/m11723/latest>

¹⁴<http://cnx.org/content/m11727/latest>

¹⁵<http://cnx.org/content/m11727/latest>

¹⁶<http://cnx.org/content/m11723/latest>

¹⁷<http://cnx.org/content/m11762/latest>

¹⁸<http://cnx.org/content/m11725/latest>

¹⁹<http://cnx.org/content/m11728/latest>

²⁰<http://cnx.org/content/m11728/latest>

²¹<http://cnx.org/content/m11723/latest>

²²<http://cnx.org/content/m11722/latest>

²³<http://cnx.org/content/m11721/latest/run.m>

²⁴<http://cnx.org/content/m11721/latest/constbuilder.m>

- [a2d.m](#)²⁵
- [constmap.m](#)²⁶
- [series2parallel.m](#)²⁷
- [mirror.m](#)²⁸
- [cyclicpad.m](#)²⁹
- [matlaberr.m](#)³⁰
- [parallel2series.m](#)³¹
- [channel.m](#)³²
- [decyclicpad.m](#)³³
- [filterchannel.m](#)³⁴
- [demirror.m](#)³⁵
- [approximate.m](#)³⁶
- [deconstmap.m](#)³⁷
- [d2a.m](#)³⁸

[Home](#)³⁹ | [Previous: Introduction](#)⁴⁰ | [Next: Results & Conclusions](#)⁴¹

²⁵<http://cnx.org/content/m11721/latest/a2d.m>

²⁶<http://cnx.org/content/m11721/latest/constmap.m>

²⁷<http://cnx.org/content/m11721/latest/series2parallel.m>

²⁸<http://cnx.org/content/m11721/latest/mirror.m>

²⁹<http://cnx.org/content/m11721/latest/cyclicpad.m>

³⁰<http://cnx.org/content/m11721/latest/matlaberr.m>

³¹<http://cnx.org/content/m11721/latest/parallel2series.m>

³²<http://cnx.org/content/m11721/latest/channel.m>

³³<http://cnx.org/content/m11721/latest/decyclicpad.m>

³⁴<http://cnx.org/content/m11721/latest/filterchannel.m>

³⁵<http://cnx.org/content/m11721/latest/demirror.m>

³⁶<http://cnx.org/content/m11721/latest/approximate.m>

³⁷<http://cnx.org/content/m11721/latest/deconstmap.m>

³⁸<http://cnx.org/content/m11721/latest/d2a.m>

³⁹<http://cnx.org/content/m11710/latest>

⁴⁰<http://cnx.org/content/m11710/latest>

⁴¹<http://cnx.org/content/m11729/latest>