

Praktikum 6

OFDM Datenpfad Transmitter

Abgabe: Bis 14.12.2017 23:59 Uhr (per e-mail). Die Praktikumsaufgaben sollten möglichst in Gruppen von 2 bis 3 Personen bearbeitet werden, damit nur eine Abgabe erfolgen muss. Die Simulationen haben mit Matlab Simulink 2016a zu erfolgen. Bei ausreichender Bearbeitung der Aufgaben gibt es 1 Punkt. Beachten Sie auch die aktuellen Hinweise auf der Vorlesungswebsite unter: <http://www.informatik.hu-berlin.de/forschung/gebiete/bbk>

Aufgabe 1

Nutzen Sie als Grundlage Ihrer Lösung von *Praktikum 5* um eine neue Simulationsdatei namens *Praktikum06.slx* zu erstellen. Erzeugen Sie ein binäres Datensignal mit einem *Bernoulli Binary Generator* Block und verwenden Sie den Modulator und Faltungskodierer aus *Praktikum 3*, die in einen Subsystem namens *Code and Modulate* vereinigt sein sollen. Kopieren Sie auch die *Modell Parameter* Maske dazu und überprüfen Sie die Funktionsweise seiner Variablen. Erzeugen Sie zunächst einen Subsystem Block namens *OFDM-Transmitter*, der ein OFDM-Frame mit der gleichen Struktur wie in *Praktikum 5* generieren soll. Ordnen Sie zuerst die QAM-modulierten Daten-Symbolen in eine $[48 \times 20]$ Matriz. Definieren Sie die Piloten mit einem *Constant* Block als einen $[1 \times 20]$ Zeilen-Vektor aus konstanten Werten $(1 + 0j)$. Fügen Sie zunächst die Nullen, Piloten und DC-Unterträger Zeile-Vektoren an den jeweiligen Positionen innerhalb des OFDM-Frames ein, so dass Sie eine $[64 \times 20]$ Matriz bekommen (*Matrix Concatenate* Block). Verschieben Sie analog zu *Praktikum 4* diese Matriz für die IFFT (Abb.1) und normalisieren Sie den IFFT-Ausgang durch eine Multiplikation mit $64/\sqrt{48}$. Ordnen Sie das IFFT-Ausgangssignal in einem Spalten-Vektor durch einen *Reshape* Block an.

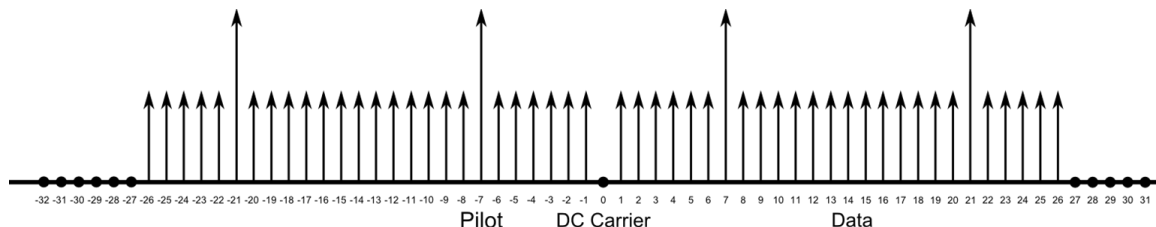


Abbildung 1: OFDM-Symbol

Erstellen Sie ein Subsystem namens *Preamble* innerhalb des *OFDM-Transmitters*, um die Präambel des OFDM-Frames zu generieren. Da die Präambel konstant ist, soll diese auch durch einen *Constant* Block als ein $[53 \times 1]$ Spalten-Vektor mit jeweils den Werten $(1 + 0j)$ definiert werden. Multiplizieren Sie diesen Vektor zu einer $[53 \times 4]$ Matriz durch einen *Selector* Block. Füllen Sie diese Matriz mit Nullen auf eine Größe von $[64 \times 4]$ auf und führen Sie, wie oben, die Verschiebung, IFFT-Bildung und Normierung durch. Ordnen Sie das IFFT-Ausgangssignal in gleicher Weise wie oben in einem Spalten-Vektor und verwenden Sie einen *Concatenate* Block, um die Präambel mit dem OFDM-Datenpfad zu einem einzelnen Spalten-Vektor verketteten.

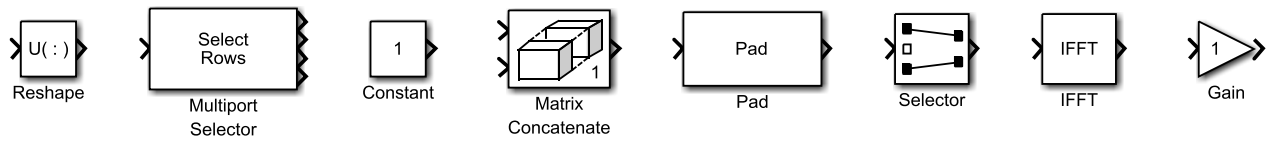
Aufgabe 2

Kopieren Sie die Subsysteme *BBK-Kanal*, *OFDM-Receiver* und *Decode and Demodulate* aus *Praktikum 5*, um die Funktionsweise Ihres *OFDM-Transmitters* damit zu testen. Führen Sie eine Simulation aus und berechnen Sie die Bitfehlerrate für die folgenden Parameter:

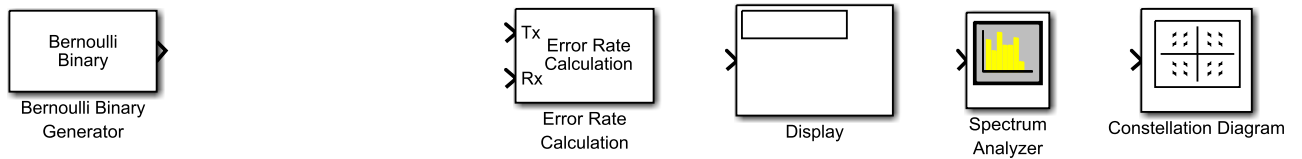
- Model Abtastrate: 20 MHz, Simulationsdauer: 10 ms
- Kanal: nur AWGN-Rauschen mit $SNR = 15$ dB, *EnableMultipath* = 0
- Demodulator: $M = 16$ (QAM), Average Power, Approximate log-likelihood ratio
- Viterbi Decoder: *Trellis* = `poly2trellis(7,[171 133])`, *TracebackDepth* = 48, *PunctureVector* = `[1; 1; 0; 1; 0; 1]`

Zusatzblatt zum Praktikum 6 – Nützliche Simulink Blöcke

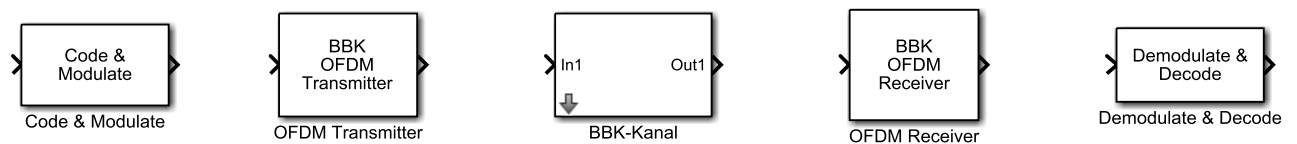
Verschiedene Blöcke



Quellen und Senke



Übersicht der Subsysteme in Top-View des Modells



Parameter Berechnung

Bernoulli Binary Generator:

Sample Time: $T_s / (48 \cdot 20 \cdot \log_2(M) \cdot \text{CodeRate}) \cdot 64 \cdot 24$

Samples per Frame: $48 \cdot 20 \cdot \log_2(M) \cdot \text{CodeRate}$