TU Berlin Fakultät IV
Institut für Telekommunikationssysteme
Fachgebiet Nachrichtenübertragung
Praktikum Nachrichtenübertragung

Praktikum 06
Digitale bertragungstechnik: Digitale Empfnger

Özgü Dogan (326 048)
Boris Henckell (325 779)

6. Juli 2012

Gruppe: D03

Inhaltsverzeichnis

1 Einleitung **1**

2 Motivation **1**

3 Theorie **1**

4 Vorbereitungsaufgabe **1**

5 Labordurchführung **2**
5.1 Aufgabe 2.1 - Aufbau des Versuches 2
5.2 Aufgabe 2.2 - Bitfehlermessung 3

6 Auswertung **3**
6.1 Vorbereitungsaufgabe 3
6.2 Aufgabe 2.1 - Aufbau des Versuches 3
6.3 Aufgabe 2.2 - Bitfehlermessung 4

7 Zusammenfassung **4**

1 Einleitung

TODO:

Einleitung schreiben

2 Motivation

3 Theorie

4 Vorbereitungsaufgabe

Zunächst sollten wir uns als Vorbereitung für den Praktikumstermin mit dem Inhalt des Kapitels zu optimalen Empfängerstrukturen aus dem Skript vertraut machen und nachvollziehen können.

Als nächstes sollte mit Hilfe einer Matlab Datei bipolare Sendeimpulse für eine bipolare Übertragung definiert werden. Diese Sendeimpulse sollten als Vektoren SF0 und SF1 mit -1 und 1 ein Kreuzkorrelationsergebnis von $\rho = 0, -\frac{1}{3}$ oder -1 liefern, wobei die Vektoren für $\rho = -1$ und $\rho = -\frac{1}{3}$ drei Werte und für $\rho = 0$ vier Werte besitzen müssen.
Als Kontrolle dafür, sollte in der Matlab-Datei tatsächlich die Kreuzkorrelation durchgeführt und das Ergebnis untersucht werden. Außerdem wird die Bitenergie E_B der Sendeimpulse berechnet, wobei die Vektoren als zeitkontinuierliche Spannungsverläufe der Dauer T_B angenommen wird, bei denen Die Spannungsamplituden nur $+1\ V$ oder $-1\ V$ betragen können. Es sollte beachtet werden, dass $T_{SF} = 20\mu s$ konstant bleiben und sich somit das T_B mit $T_B = 2 \cdot N_{\cdot SF}$ mit N Werten in SF0 bzw. SF1 berechnen lässt. Als Bedingung sollte noch gelten, dass die Bitenergien für SF0 und SF1 als Paar für ein ρ gleich sein sollten.

Als Vorbereitung für den praktischen Teil des Versuchs, sollte eine Funktion SAF implementiert werden, welche einen SAF-Empfänger in Matlab realiseren soll. Der Input besteht aus DataSamples, ClkSamples und SFSamples, welche gegeben sind. der Output dagegen soll dabei der Vektor Values sein, in der die Werte nach dem signalangepasstem Filter eingetragen sein sollen.
TODO:

Boris: kannst du hier noch genauer erklären wie die Funktion funzt?

Nach der Implementierung der SAF-Funktion sollten wir uns mit dem Versuchsaufbau und der Durchführung vertraut machen und ein Blockschaltbild dazu entwerfen. Die genaue Vorgehensweise ist in dem Abschnitt Durchführung detaliert erläutert. Hier ist das entworfene Blockschaltbild zu dem Versuch:

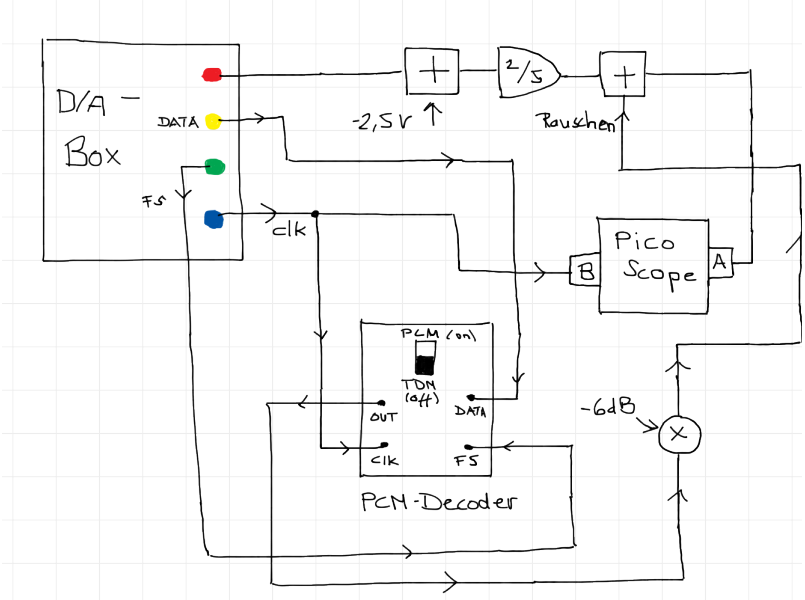

Abbildung 1: Blockschaltbild des Versuchsaufbau

Um die implementierte Funktion SAF zu überprüfen, führten wir mithilfe der vorgegebenen ENue_SAF_Messumgebung.m durch, welche das Hauptprogramm der Messung darstellt. Mit dem Parameter Simulation (für die Vorbereitung bleibt dieser bei 1) kann man entscheiden, ob die Ausgabe und Einstellungen des Steckbretts simuliert werden, oder ob die Daten über die D/A-Box ausgegeben und per PicoScope eingelesen werden. Mit dem Parameter SAF kann man einstellen, ob mit dem DataSignal eine Sendeformung und dem Empfangssignal eine signalangepasste Filterung durchgeführt wird. Weiterhin sollten wir für die Vorbereitung die Kanal- und Filtereinstellungen nicht verändern, wohingegen der NoiseFaktor, welcher den Faktor des Rauschens einstellt, variiert werden durfte um die Leistung AWGN (additive white Gaussain noise) des Kanals zu beeinflussen.
Mit der Funktion Channel führt man die Übertragung entweder als Simulation oder auf dem ETT aus, wobei die Einstellungen der Simulation denen des wirklichen Aufbaus entsprechen. Es wird ein Rückgabevektor Y ausgegeben, welcher die Werte nach der Abtastung des Kanals enthält. Wenn SAF deaktiviert ist, wird eine einfache Nachabtastung mit einem Schwellwert von $0V$ durchgeführt.
Außerdem entspricht der Rückgabewert Noise einer Rauschmessung des Kanals, wozu eine 0-Bit-Folge gesendet und das Empfangssignal ohne SAF oder Nachabtastung als PuciScope-Sample-Signal zurückgegeben wird. Dieses Signal enthält ungefähr 10000 Samples.

Zuletzt sollte beantwortet werden, wie der Multiplikationsfaktor für das Rauschen variiert werden müsste, damit die Wasserfallkurve ausreichend ermittelt werden kann. Im Versuch beträgt das Rauschen dabei $-6dB$.

5 Labordurchführung

5.1 Aufgabe 2.1 - Aufbau des Versuches

Die erste Aufgabe des Praktikums beschäftigt sich fast ausschließlich mit dem Aufbar des Versuches, um den Signalverlauf mit der Matlab Funktion ParallelOUT($[0101010], 100$) zu untersuchen. Dazu wird die D/A-Box, welcher an den Computer angeschlossen und somit über die Matlab Dateien steuerbar ist,

als Quelle für jegliche verwendete Signale verwendet. Die D/A-Box besitzt vier Ausgänge, die mit unterschiedlichen Farben gekennzeichnet sind. Der rote Ausgang gibt das DataSignal aus. Dieser kann eine Amplitude von $0V$ oder $5V$ besitzen. Um ihn auf den von uns erwünschte Spannungsbereich von $\pm 1V$ zu bringen, wird dieses Signal zunächst mit einer Gleichspannung von $-2.5V$ aus der variablen Spannungsquelle des Steckbretts addiert und danach mit einem Faktor von $\frac{2}{5}$ gedämpft. Damit erreichen wir den nötigen Spannungsbereich, welcher auf dem A Kanal des PicoScopes kontrolliert wird.
Der blaue Ausgang der D/A-Box gibt das Clock-Signal wieder. Dieser wird auf den B Kanal des PicoScope geführt und ebenfalls kontrolliert.
Als nächstes wird der PCM Decoder mit der D/A-Box verbunden, indem das Clock-Signal auch auf den Clock-Eingang des Decoder geführt wird. Weiterhin wird das Signal aus dem grünen Ausgang der Box, welcher das Frame-Signal (FS) wiedergibt, mit dem FS-Eingang des Decoders und die PCM-codierten Datenworte, die aus dem gelben Ausgang der Box entnommen werden können, mit dem PCM Data-Eingang des Decoders kontaktiert. Wichtig ist auch, dass der Schalter auf dem Decoder Modul auf PCM geschaltet ist und nicht auf TDM.
Somit ist der PCM Decoder mit allen Signalen beliefert, die er zum decodieren braucht. Daher kann man nun das Output Signal verwenden, welcher eine Spannung des Faktors wiedergibt, die für die Verstärkung oder Dämpfung des Rauschens dient. Diese Spannung wird an einem Multiplikator mit dem $-6dB$ Rauschen multipliziert und an dem Addierer mit Multiplikatoren mit dem DataSignal, welcher bereits auf den korrekten Spannungsbereich eingestellt wurde, addiert. Das Ergebniss dieser Addition wird weiterhin auf den A Kanal des PicosScopes geführt und ausgewertet.

Zuletzt wird das verrauschte Signal überprüft, indem mit der Funktion PCM_Decod(192) ein Verstärkungsfaktor für das Rauschen gesetzt wird. Das verrauschte Signal wird mithilfe der PicoScope-Software und der Matlab-Funktion ParallelOUT($[0101010], 100$) dargestellt und auch mit den Werten 0, 128 und 255 untersucht.

TODO:

Foto vom Aufbau einfügen

5.2 Aufgabe 2.2 - Bitfehlermessung

TODO:

Durchführung zur Bitfehlermessung einfügen

6 Auswertung

6.1 Vorbereitungsaufgabe

6.2 Aufgabe 2.1 - Aufbau des Versuches

6.3 Aufgabe 2.2 - Bitfehlermessung

TODO:

Aufgabe 2.2

7 Zusammenfassung

TODO:

Zusammenfassug schreiben

Literatur