

UHD を用いたソフトウェア無線 OFDM の実装について

山崎 景太[†] 石橋 功至[‡] 猿渡 俊介[†] 渡辺 尚^{‡†}
[†] 静岡大学 [‡] 電気通信大学 ^{‡†} 大阪大学

1 はじめに

筆者らは、無線 LAN 技術を拡張して、無線電力伝送や電波による空間のセンシングなど次世代情報通信技術の新たな基盤を実現することを目指している。電力伝送やセンシングなどの多様な領域が複雑に絡み合う研究開発を円滑に進めるためには、理論構築から実証までを迅速に遂行するための仕組みが求められる。本稿では、現在の無線 LAN 技術で標準的に利用されている OFDM のソフトウェア無線実験システムについて述べる。本システムは全て C++ を用いて実装されているため、C 言語の知識を有するユーザであればシミュレーションから実機計測までをシームレスに実現できる。

2 関連研究

全ての無線通信の処理を PC 上で実現できるミドルウェアとして、GNU Radio が挙げられる。GNU Radio では、あらかじめ用意された信号処理ブロックをスクリプト言語である Python を用いて接続するだけで簡単に無線通信機を実装できる。しかしながら、GNU Radio で OFDM のような複雑な信号処理を実現する場合には次の 2 つの問題がある。

1 つ目は、GNU Radio で採用されている入出力ブロックモデルで記述されたソフトウェアを他の研究者や開発者が利用する場合に読解が困難であるという問題である。複数の入出力ブロックを接続して 1 つの処理を完成させるというプログラム形態は初学者にとっては直感的に理解しやすいというメリットがある。一方で、大規模な処理を記述する場合、どのブロックとどのブロックが接続されているのか、どの順番に処理が実行されるのかを理解するのが困難になるという問題が発生する。

2 つ目は、Python での実行は処理速度が遅いという問題である。GNU Radio では、Python から C++ で作成された信号処理ブロックを接続して物理層の処理をする。物理層の処理のみに限定した場合には実質的には C++ で処理しているのと変わらない性能が得られる。しかしながら、Python 自体では高速性が求められる信号処理は記述できず、信号処理を拡張する場合には結局は開発者は C++ を用いなければならない。

3 UHD を用いた OFDM 実験システム

3.1 全体像

図 1 に OFDM の実験システムの外観を示す。送信側 PC と送信側 USRP N200 では、入力された 1 次変調点を OFDM 変調して、電波として送信する。受信側 PC と受信側 USRP N200 では、送信側が送信した電波を受信したものを OFDM 復調して信号点を出力する。送信側 USRP N200 と受信側 USRP N200 は、減衰器を介して有線で接続した。Fury Desktop は送信側の USRP N200 と受信側の USRP N200 に接続した。Fury Desktop から出力される 10 [MHz] の信号を送信側と受信側の USRP の両方に入力することで、USRP N200 間の周波数偏差が小さくなるようにしている。OFDM 変調/復調のプログラムは RawOFDM [1] を元に作成した。以下では送信機、受信機の実装についてそれぞれ詳しく述べる。

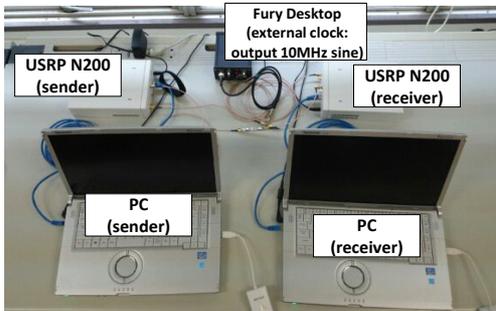


図 1: OFDM の実験システムの外観

3.2 送信機

送信機では大きく分けてプリアンブル作成、データシンボル作成、OFDM 変調、スケール調整、送信の合計 5 つの動作を行う。図 2 に 1 シンボルのサブキャリアを示す。IEEE 802.11a に倣いデータ信号用に 52 個のサブキャリアを用いる。データ信号用の 52 個のサブキャリアの周囲と中心には、ガードバンドとして合計 12 個のサブキャリアを設けている。FFT 長はガードバンドを考慮して 64 とした。

1 つ目の動作であるプリアンブル作成では、3 つの既知の OFDM シンボルを作成する。1 つ目のシンボルは、フレーム検出のためのシンボルである。受信側の自己相関でフレームの検出ができるように、26 個のサブキャリアの奇数番のみに 26 サンプルのランダムな複素数値を代入することで、IFFT 後の信号の前半と後半が一致するようにしている。ランダムな複素数値は、IQ の絶対値が $\sqrt{2}$ である既知の値を使用する。2、3 個目のシンボルは、チャネル補正のためのシンボルである。52 個のサブキャリアすべてに 52 サンプルのランダムな複素数値を代入する。ランダムな複素数値は、IQ の絶対値が 1 となる既知の値を使用する。

2 つ目の動作であるデータシンボル作成では、1 次変調された信号点を 48 サブキャリアにそれぞれ代入する。52 サブキャリアのうち残り 4 サブキャリアには BPSK のパイロットシンボルを代入する。パイロットシンボルには先頭からビット 0 とビット 1 の値が交互に代入されている。

3 つ目の動作である OFDM 変調では、作成したプリアンブルとデータシンボルにそれぞれ IFFT 処理を行うことで変調する。図 3 に IFFT 処理を行う前の信号を示す。先頭 3 シンボルにはプリアンブルが代入されており、4 シンボル以降には 16 個のデータシンボルが代入されている。この各シンボルにそれぞれ IFFT 処理を行うことで変調する。変調された信号は、周波数軸上のサブキャリアから時間軸上の信号へと変換される。図 4 に IFFT 後の信号に cyclic prefix を付加した 4 シンボルの信号を示す。シンボル間干渉を防ぐために、シンボルの最後尾から 16 サンプルを cyclic prefix としてシンボルの先頭に付加している。

4 つ目の動作であるスケール調整では、cyclic prefix を付加した信号に対してスケール調整を行う。具体的には、USRP の DA 変換器が許容するダイナミックレンジである 1 から -1 の値に信号全体が収まるように OFDM シンボルの振幅包絡線を調整する。OFDM 信号は、平均電力レベルに対してピーク電力が非常に大きい。作成した信号が USRP の許容するダイナミックレンジを超えた場合には、信号のクリップや大きな歪みが発生する。

5 つ目の動作である送信では、スケール調整済みの信号を UHD を介して USRP に送信すると、USRP から電波として送信される。

3.3 受信機

受信機では受信信号の取得、シンボル同期、位相オフセット補正、OFDM 復調、オフセット補正、シンボルから信号の取出しの 6 つの動作を行う。

1 つ目の動作である受信信号の取得では、UHD を用いて USRP で受信した電波を受信信号として受け取る。

2 つ目の動作であるシンボル同期では、フレームを検出する。図 5 にプリアンブルの 1 シンボルの受信信号と半波長遅延させた受信信号を示す。プリアンブルの 1 シンボルのみ信号の前半と後半が一致しているため、受信波形と半波長遅延させた受信信号が半波長と cyclic prefix 分、48 サンプルの間で一致することが分かる。半

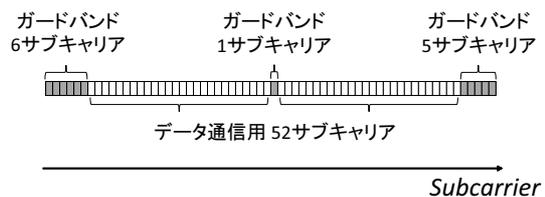


図 2: 1 シンボルのサブキャリア

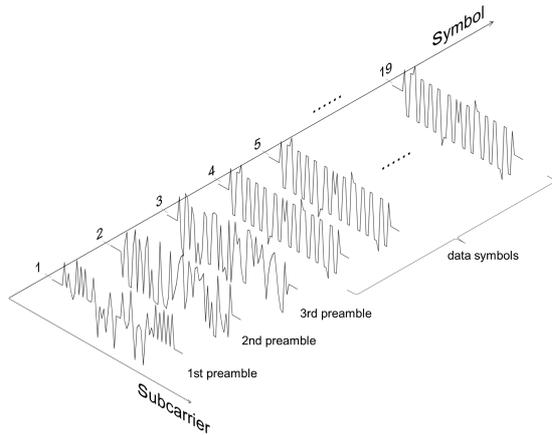


図 3: IFFT 前の信号

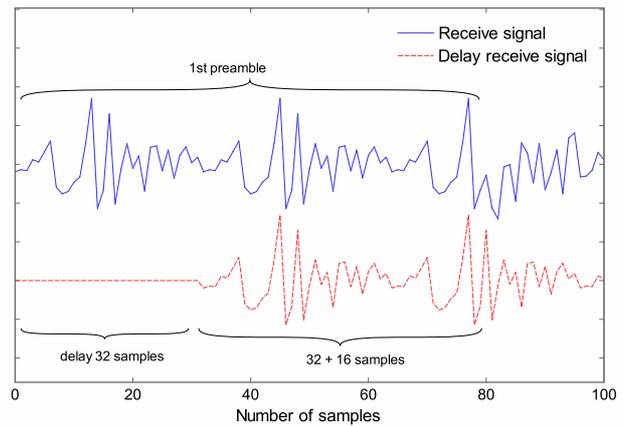


図 5: 受信信号と遅延させた受信信号

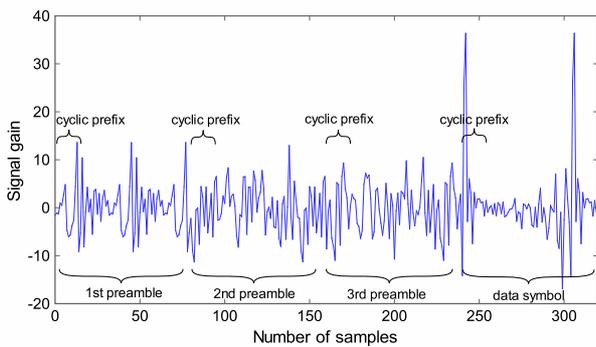


図 4: IFFT 後の信号

波長ウィンドウ幅の畳み込み積分で自己相関を算出すると、ピークは cyclic prefix 分、16 サンプルの間で立つことが予想される。しかしながら、自己相関の値は正規化が行われていないため受信開始時の AGC (Automatic Gain Control) が正しく効いていない場所では受信電力自体が膨大になりピークが立ってしまう。受信開始時の AGC が正しく効いていない部分でピークを観測しないようにするために、自己相関の値を受信電力で除算することで正規化する。正規化された相関を 16 サンプルのウィンドウ幅で移動平均を取ること、フレームの開始地点にピークが立つ。ピークの検出では、実測により得た値を閾値としてフレームの開始地点を決定した。

3 つ目の動作である周波数オフセット補正では、周波数オフセットによって時間領域で信号点が回転する現象を逆回転させて補正する。周波数オフセットの算出には、1 つ目の動作で用いた自己相関の移動平均と、2 つ目の動作で決定したフレームの開始地点を用いる。1 つ目の動作で用いた自己相関の移動平均の位相値は、受信信号と半波長遅延させた受信信号の半波長間での平均のずれと等しくなる。具体的には、受信信号と半波長遅延させた受信信号が半波長間に渡って一致している場合には位相値は 0 度と算出されるが、周波数オフセットにより信号点が回転していた場合には、半波長間に渡って回転した角度の平均が算出される。算出された角度を用いて、受信信号に対して逆回転するよう補正する。

4 つ目の動作である OFDM 復調では、オフセット補正された受信信号に対して cyclic prefix を削除した後、FFT 処理を行うことで復調する。Cyclic prefix の削除では、シンボル同期で発見したフレームの開始地点を元に、オフセット補正された受信信号のシンボルの先頭 16 サンプルを削除する。Cyclic prefix を削除した信号を FFT 処理することで、信号を時間軸上から周波数軸上のサブキャリアに変換する。

5 つ目の動作であるチャネル補正では、推定したチャネルを用いて周波数による特性の変動 (周波数選択性フェージング) を補正する。チャネル推定では、プリアンプの 2, 3 シンボル目を用いて、送信したプリアンプの値を受信したプリアンプの値で除算した値を用いる。求めたそれぞれの周波数帯に対するチャネル情報を保存しておき、フレーム受信毎にチャネル情報を更新する。求めたチャネル情報を FFT 処理後の信号に乗算することで、チャネル補正を行う。

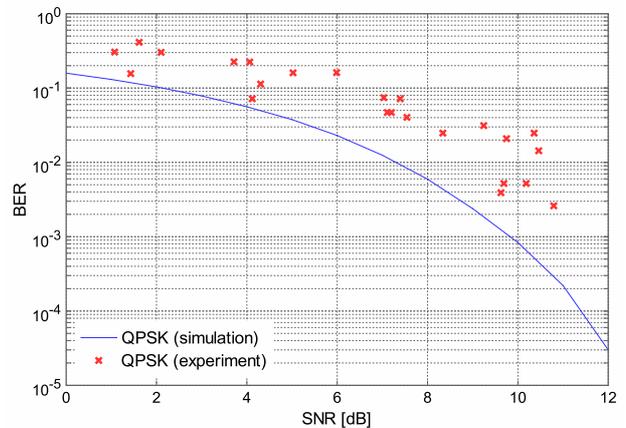


図 6: qpsk のシミュレーションと実測結果

6 つ目の動作であるシンボルからの信号の取出しでは、52 サブキャリアから信号点を取り出す。取り出された受信信号点は通信路による位相回転等の擾乱を受けているため、これをパイロットシンボルから推定した情報を用いて補正し、復号器に渡す。

4 動作検証

OFDM の実験システムの初期的評価として、シミュレーションによる値と実測値の、SNR (signal to noise ratio) に対する BER (bit error rate) を比較した。シミュレーションでは、AWGN (Additive White Gaussian Noise) 通信路の BER を示す。シミュレーション、実測共に 100000 [bits] 送信した際の BER を算出した。

図 6 に QPSK における SNR に対する BER の評価を示す。図 6 から次の 2 つのことが分かる。1 つ目は、実測による値がシミュレーションに漸近する特性を示していることである。OFDM の実験システムが正しく動作していると考えられる。2 つ目は、実環境の干渉によって特性が悪くなっていることが分かる。シミュレーションでは AWGN による擾乱のみを考えているのに対し、OFDM の実験システムによる実測では増幅器における非線形歪みやオシレータの精度といった擾乱成分もあるためだと考えられる。

5 おわりに

本稿では、ソフトウェア無線機と UHD を用いた OFDM の実験システムを実装した。現在、構築した OFDM の実験システムを用いて、無線電力伝送、逐次干渉除去、重畳符号化、レートレス符号を用いた新たな無線通信方式の研究を進めている。

謝辞

本研究は総務省・2013 年度戦略的情報通信研究開発推進事業 (SCOPE) の助成を受けて行った。

参考文献

- [1] How to build and run RawOFDM:
<http://people.csail.mit.edu/szym/rawofdm/>.