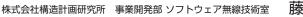
ソフトウェア無線(SDR)技術の最新動向 と将来展望

ふじい よしみ 藤井 義巳





1. はじめに

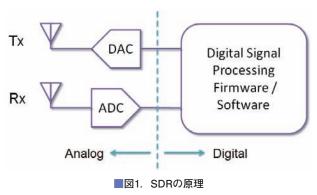
弊社のSDRへの取組みは、2013年初頭にドイツのFraunhofer Heinlich Hertz研究所(Fraunhofer HHI)を訪問した時から始まる。Fraunhofer HHIは当時既に5Gに向けた研究試作プラットフォームとしてSDRを活用していた。私は帰国後すぐにSDRの事業化に向け、国内におけるSDRの普及に向けて活動を開始した。それから4年、2017年現在まだまだ国内でSDRの利用が十分に進んだとは言えない。本稿は我々の経験から、現時点で日本国内においてSDRを取り巻く環境と、今後どのようにしてSDRを世の中に浸透させ、発展させていくかという観点から筆を進める。

2. SDRとは何なのか?

2.1 ソフトウェア無線の原理

そもそもソフトウェア無線(SDR: Software Defined Radio)とは何なのか。当初英語ではSoftware Radio(SR)と呼ばれていたこの技術は、他の多くの分野同様、アナログ電子回路で実現されていた部分をデジタル回路によって置き換え、さらにその延長上でデジタル回路と等価なコンピュータソフトウェアがそれを置き換えるという自然な流れを、無線(通信)の分野に適用した概念である。(受信機の場合)対象とする周波数の高周波信号を高速なA-Dコンバータでデジタル化することができたとして、十分に高速な計算機を用意することができれば、どのような無線方式であろうとソフトウェアによるデジタル信号処理にて実現することが可能になるという理屈である。

この事に異議を唱える人はいないだろう。しかし、その 実現には当初思った以上の年月を要している。



原理的にSDRというものが可能になったからといって、直ちに従来の「ハードウェア」無線機がすべてソフトウェア無線機に置き換えられる訳ではない。NTTの研究所でソフトウェア無線機のプロトタイプを私が初めて目にした1990年代後半から既に20年が経過したが、ソフトウェア無線機がハードウェア無線機を駆逐してしまったという話は未だ聞かない。むしろその逆で、ソフトウェア無線の普及は2017年になった今でもまだ道半ばという状況である。では何がその妨げになっているのか。SDRを普及させるには、何がその妨げになっているのかをきちんと見極め、一つ一つ取り除いていく必要がある。

2.2 ソフトウェア無線の定義 (その1)

私の経験上、ソフトウェア無線という言葉の定義は使う人によって異なる。つまり、「ソフトウェア無線」あるいは「SDR」と口にするとき、同じ単語でありながら、それらが意味するところは千差万別である。その人の立場によって、あるいはその人の理解の度合いによって、SDRが意味するところは全く異なってくるのだ。例えば「ソフトウェア・・・」と言いながら、おそらく多くの技術者がすぐに思い浮かべるのは、RTL-SDRであったりUSRPであったりする。だが、私に言わせればそれは「ハードウェア」であって、「ソフトウェア」ではない。

ソフトウェア無線はあくまでも「ソフトウェア」が主人公であるべきであり、本稿ではUSRPのようなハードウェアは「SDRデバイス」と呼ばせていただく。「ソフトウェア無線」はソフトウェア技術者が無線の技術を手にするための技術に他ならない。その意味でも「ソフトウェア無線」がRTL-SDRやUSRPなどのハードウェアを指し示す言葉であってはならないのである。

Wikipedia (日本語) によるソフトウェア無線の定義は以下のとおりである。重要と思われる箇所に下線を引いている。

ソフトウェア無線(Software-defined radio)とは、電子回路(ハードウェア)に変更を加えることなく、制御ソフトウェアを変更することによって、無線通信方式を切り替えることが可能な無線通信、又はその技術。一般的には、広い周波数範囲において多くの変調方式が

可能となるよう、ソフトウェアが、なるべく汎用性の高いプログラム可能なハードウェアを制御するものとして考えられている。

2.3 ソフトウェア無線の定義 (その2)

近年、いろいろな場面で「ソフトウェア無線機」あるいは「DSP無線機」と呼ばれる、内部の信号処理をソフトウェアで実現した無線機を見るようになった。例えばアマチュア無線。近年発売された比較的高級なアマチュア無線機の多くは「DSP」が当たり前となってきた。海外メーカーからは「フルSDR無線機」と称するものも発売されている。しかし本当にそれらを「ソフトウェア無線機」と呼んでいいのか。私の定義では、それらは「ソフトウェア無線の技術を使って開発された無線機」ではあっても、私の考える「ソフトウェア無線機」ではない。ソフトウェア技術者がソフトウェアを駆使して課題解決に使える道具、それが「ソフトウェアを駆使して課題解決に使える道具、それが「ソフトウェア無線機」に他ならない。ソフトウェア無線は、下記3つの点を満たすようなハードウェアとソフトウェア開発環境を兼ね備えたものでなければならないと私は考えている。

●幅広い周波数に対応可能

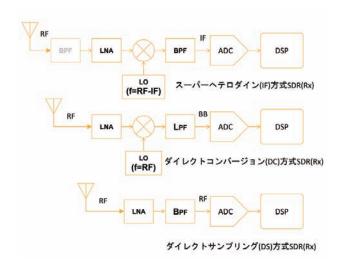
- ・SDRデバイスの内部には、なるべく周波数範囲を制約 するハードウェアを持たないことが望ましい。それら は必要に応じて外部接続すればよい。
- ●ソフトウェア制御可能な汎用のハードウェア
 - ・ハードウェアは特注品ではなく、(なるべく安価な) 汎用の市販品であること。また、ソフトウェアから制御 するためのデバイスドライバやAPIを標準で備えてい ること。それらはオープンソースであることが望ましい。
- ●ソフトウェアにより変復調などのデジタル信号処理が可能
 - ・PCあるいはDSP、FPGAを用いて自由にソフトウェア の作成が可能であること。

これら3点を満たして初めて、SDRはソフトウェア技術者が使える道具となり得るのである。本稿では以後、このような視点から、ソフトウェア無線の世界を取り巻く環境の現状と、これからの展望について見ていきたい。

3. SDRの現在

3.1 ソフトウェア無線の方式

この20年ほどの間に多くのソフトウェア無線方式が考案されてきた。AD/DA変換の前後にRF信号の周波数変換を行うかどうか、行う場合にどのような変換を行うかで図2に示す3つの方式に分類できる。



■図2. 代表的なソフトウェア無線方式(受信機)

3.1.1 スーパーヘテロダイン (IF) 方式

スーパーへテロダイン方式は、隣接周波数からのイメージの発生を避けるためにRF周波数帯でバンドパスフィルタ (BPF) が必要になるという欠点を除けば非常に優れた方式である。BPFの通過帯域の周波数を可変させようとするとき、多くの場合にはLC共振回路が用いられる。残念ながらSDRに要求される広い周波数をLCで実装されたBPFでカバーしようとすると、多くのLC回路を切替えて使うより他に方法が無く、このことがSDRデバイスにスーパーへテロダイン方式の採用が避けられる最大の理由となっている。ただし、狭帯域の無線をソフトウェア無線で実現する場合には、この方式が採用される傾向にある。

3.1.2 ダイレクトコンバージョン (DC、ゼロIF) 方式

ダイレクトコンバージョン方式はゼロIF方式とも呼ばれ、スーパーヘテロダイン方式で中間周波数を0Hzにした方式であるというとらえ方もできる。原理的にイメージの問題が発生しないことから、周波数変換前のBPFも省略することが可能である。また、IFを0Hzにすることで、周波数変換後のフィルタにBPFではなくLPFを採用することができるため、回路が単純化できるメリットもあり、携帯電話や無線LANの子機など小型化の要求からLC回路を搭載しにくい無線機器にはこのDC方式が広く使われるようになった。現在広く普及しているSDRデバイスの多くがこの方式を採用している。

3.1.3 ダイレクトサンプリング (DS) 方式

サンプリング定理によると、A-D変換前のアナログ信号 にサンプリング周波数f₆の1/2を超える周波数成分が含ま れていると、エイリアス成分となってサンプリング結果に



混入し、歪の原因となる。そのことから、スーパーへテロダイン方式にしてもダイレクトコンバージョン方式にしても、周波数変換後にBPFあるいはLPFを用いて帯域外成分を除去するのが当たり前となっている。このダイレクトサンプリング方式は、このエイリアス成分を積極的に利用することで、周波数変換を経ることなく直接RF信号からデジタルサンプルを得る方式である。LOの位相雑音が周波数変換により混入されることを避けられるなどのメリットがあり、一部のSDRではこの方式が採用されている。

3.2 様々なSDRデバイス

2017年現在、多くのベンダーから非常に多種多様なSDR デバイスが入手可能である。ここでは比較的容易に入手可能な安価(数千円~数万円)なSDRデバイスに絞って一覧表で比較してみた。対応する周波数範囲やPCとのI/Fの違い、チャネル数などの違いはあるものの、受信専用のRTL-SDRを除けばどのデバイスもスペックに大きな差は見られない。2017年になってから、教育用途を意図した1万円台のSDRデバイスが登場し、SDRの普及に拍車がかかることが期待される。

3.3 主要なソフトウェア開発環境

SDRはソフトウェアが要となることを先に述べた。汎用

のソフトウェア開発環境でも開発は可能だが、デジタル信号処理は汎用のプログラミング言語とソフトウェア開発環境ではなかなか効率良く開発できない。そこで、SDRに適したソフトウェア開発環境がいくつか提案されている。

3.3.1 GnuRadio

GnuRadioという名称からも分かるように、オープンソースのSDR専用開発環境である。当初、USRPのソフトウェア開発環境として、ベンダーであるEttus Research社が多大な貢献をして開発された。GitHub上にソースコードが公開されていて、多くのSDRプログラマーからのコントリビューションが寄せられている。

個々の信号処理のコンポーネントはC++により開発され、それらをPythonでつなぎ合わせて使用する。GnuRadio Companion(GRC)と呼ばれるGUIを使えばグラフィカルなプログラミングにより様々な信号処理を実現可能である。GnuRadioは当初Linux上で開発されたが、現在、Windows用のインストーラも用意されている。

3.3.2 LabView

LabViewは、米国のNational Instruments社が販売する 実験機器の制御や信号処理を可能にするビジュアルプロ グラミング環境である。SDR専用というわけではないが、

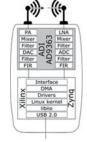
	RTL-SDR	USRP	bladeRF	LimeSDR	ADALM-PLUTO
ベンダー	RTLSDR.comなど	米Ettus Research, Inc.	米Nuand, Inc.	英Lime Microsystems, Ltc.	米Analog Devices, Inc.
用途	ホビー、研究	研究、試作、教育	研究、試作、教育	研究、試作、教育	教育
周波数	28MHz ~ 1700MHz	70MHz ∼ 6GHz	300MHz ~ 3.8GHz	100kHz ∼ 3.8GHz	70MHz ∼ 3.8GHz
帯域幅	2.8MHz	56MHz	28MHz	56MHz	20MHz
A-D/D-A	8bit	12bit	12bit	12bit	12bit
チャネル数	1 (RXのみ)	1 or 2	1	2	1
PCとのI/F	USB2.0	USB3.0 or Ethernet	USB3.0	USB3.0 or PCle	USB2.0
FPGA搭載	無	有、機種により異なる	Cyclone IV	Cyclone IV	ZYNQ7100
価格	1,000円~3,000円	8万円~数十万円	5.4万円~ 8.4万円	3万円~	1.5万円
備考	IF方式	DCまたはDS方式	DC方式	DC方式	DC方式

■表. 代表的な市販SDRデバイスの比較

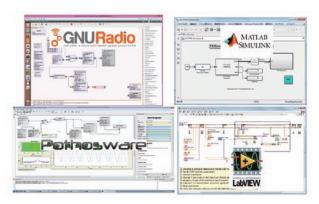


■図3. bladeRF





■図4. ADALM-PLUTO



■図5. 様々なSDRソフトウェア開発環境

NI社はEttus Research社をはじめとするSDR関連企業を次々と傘下に収め、自社のSDRデバイス向けにLabViewをその開発環境として提供してきている。ビジュアルプログラミングだけでなく、最終的にFPGAやSoCなど組込み環境で動作するSDRコードを生成するところまでサポートされている。

残念なのはNI社以外のSDRデバイスにLabViewが対応 していないことである。

3.3.3 MATLAB / Simulink

よく知られている米国のMathworks社の汎用シミュレーションプラットフォームMATLAB / SimulinkもSDRのソフトウェア開発環境としてよく使われている。Mathworks社は自社のSDRデバイス製品を持っていないため、比較的ニュートラルな立場で多くのSDRデバイスとのI/Fがサポートされている。特に米国のAnalog Devices社とは緊密な関係にあり、同社のSDRデバイスについてはMathworks社純正のハードウェアサポートパッケージとしてI/Fコードだけでなく様々なサンプルプログラムと共に提供を行っている。

MATLAB / SimulinkもLabView同様、FPGAなどの組込み環境で動作するコードの生成まで含めたサポートが充実している。

3.3.4 その他

上記3つ以外にも、PothosSDRといったフリーの開発環境も提案されている。

また、FPGAベンダーはSDRを意識した開発ツールとそこで利用可能なIP coreを揃えて提供している。例えば、米国のIntel(旧Altera)が提供するDSP BuilderはSimulinkにインテグレート可能な開発環境であり、Simulink上でのシミュレーションによる検証からHDL生成までシームレスに行うことが可能である。Mathworks社のHDL Coderも同様の目的で提供されている。最終的にリアルタイムに信

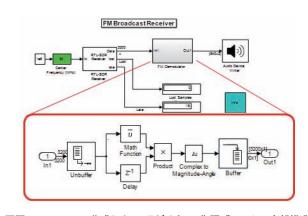
号処理を行う組込みシステムとして完成させる必要がある ため、このようなHDL高位合成の開発環境が今後主流に なるのは当然の流れと言えよう。

4. SDR応用例

現状のSDRデバイスを使って何ができるのか、どこまでできるのか、経営者であっても技術者であってもそれが最大の関心事と思われる。我々 SDRを普及させる側の立場でも同様である。SDRでできることを広く知ってもらい、またそのレパートリーを増やしていくことこそSDRの普及に直結する。

4.1 FMラジオ

従来のアナログ回路を使った変調、復調回路とは異なり、デジタル方式の良いところは、その方式の原理をそのまま素直に実装すれば良い点である。図6に示すのはFMラジオである。FM検波回路は構成する電子部品の特性について深く理解していないとその動作原理を理解することは難しい。それに対して、デジタル信号処理は数式をそのままブロックに置き換えるだけで簡単に動作する。図に示したFM復調回路はI-Q信号の複素共役をとり、1サンプル遅延させたものと乗算し、その位相角を取り出すという実にシンプルなやり方でFMベースバンド信号から音声を取り出している。このようなシンプルさは学生に無線通信の原理を教える際にも非常に有用である。

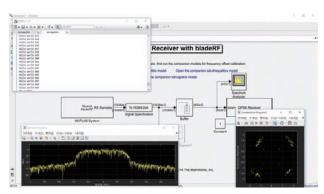


■図6. Simulinkで作成したFMラジオとFM復調ブロックの内部構造

4.2 狭帯域デジタル無線 (QPSK) 通信機

普及しているDCタイプのSDRデバイスは狭帯域の通信機を実現するのにはあまり適していないと言われているが、ごく初期段階のプロトタイピングには有用である。図7はMATLAB/Simulinkで作成した狭帯域無線機(QPSK)の例である。この例は受信機であり、Mathworks社がサ





■図7. Simulinkで実装したSDR QPSK受信機

ンプルとして提供しているものを改造している。数十kbps 程度のデータレートであるが、特に受信機側は周波数オフセット補正、シンボルタイミングリカバリといった処理を全てソフトウェアで行うとかなり処理が重くなり、PCの汎用プロセッサではリアルタイム信号処理がぎりぎりできるかどうかというレベルである。

4.3 GNSS (GPS) 信号の合成

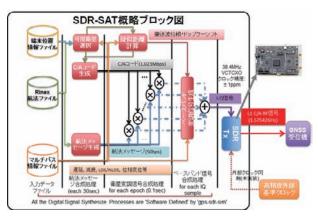
GNSS (GPS) 信号はデータレートが非常に低いことから、ソフトウェア無線のターゲットとしてはうってつけである。現在、日本上空ではGPSだけでも常に10機前後が受信可能であり、それに準天頂衛星(QZSS)を加えた衛星数分の信号をC言語で書かれた信号生成プログラムで合成する。サンプルレートにもよるが、その程度の衛星数であればCore i7 7700 (4コア4スレッド L3キャッシュ 8MB) ぐらいのCPUなら容易にリアルタイム信号生成してくれる。

マルチパスの模擬には衛星数が2倍以上になったのと同じ計算量が発生するため、GPGPUの助けを借りなければならないが、それでも従来のハードウェアシグナルジェネレータと比べればはるかに低コストで同様のことができるのは驚異的である。

5. SDRの未来

SDRデバイスは今後、他のハードウェアデバイスがそうであったように、小型化、軽量化そして低価格化していく。デジタルオーディオデバイスがPCに標準搭載されるようになったのと同様、SDRデバイスもほぼ同じ道をたどると予想される。ただしオーディオデバイスと異なり、アンテナの問題と電波法の問題をどうクリアするかという点がまだ不透明である。

SDRのデジタル信号処理をPCのメインCPUが担えるほ



■図8. SDRを使ったGNSS (GPS) 信号の合成

ど処理性能が向上するのか、それともFPGAやGPUなどの ヘテロジニアスなマルチプロセッサ構成が今後のPCの主流となるのかは分からない。しかしSDRがPCに標準搭載 のデバイスになった場合、SDRデバイス(アナログ回路 + A-D/D-A変換、デジタルフィルタなどの共通部分)の抽象化とFPGAなどの信号処理演算ブロックの抽象化が重要 なカギになると思われる。SDRアプリケーションはハードウェアの細かな違いを気にせずに作成可能となり、異なる ベンダーのSDRデバイスやFPGAを搭載していても、同じ アプリケーションが動作する環境が登場してくるだろう。

一旦OSレベルでのハードウェアの抽象化に成功すれば、あとは簡単である。ハードウェアのスケールアップ、スケールダウンにより、超小型の組込みシステムから大型の基地局に至るまで、同じフレームワークの上でソフトウェア開発が可能となる。無線機開発のかなりの部分はソフトウェア開発が占め、汎用のコンポーネントを組み合わせて作ったハードウェアに開発したSDRソフトウェアを載せて無線機が完成という時代が意外に早くやってくるかもしれない。

6. おわりに

本稿ではソフトウェア技術者の立場から見たソフトウェア無線の現状と、その将来性について述べた。現在普及しつつある市販品のSDRデバイスについて触れ、ソフトウェア開発環境、SDRを用いた応用例についても紹介した。SDRの普及を可能にしたチップベンダーの動向などについても紹介したかったが誌面の都合上割愛した。次の機会に譲りたい。

(2017年7月14日 情報通信研究会より)