On-Demand Waveform Design for Software Defined Radio Applications

Gary Ray

Abstract - The exponential growth in demand for wireless communications, both for civilian and military usage, has created an urgent need to use as much of the increasingly scarce available RF bandwidth as possible. This available bandwidth is usually non-contiguous, time-varying and non-uniform in propagation characteristics. Software defined radio (SDR) technology that implements many existing communication signals and protocols traditionally attempts to utilize available spectrum by attempting to match one or more existing modulation types to the current spectrum conditions. This can often be a poor match with serious bandwidth under-utilization. This paper describes an architecture and algorithm set for a new on-demand waveform design approach to SDR to address this problem in a wide variety of application areas, as well as an implementation using existing software defined radio hardware at Boeing.

Index Terms - Software Defined Radio (SDR), modulation, optimization, waveform design.

I. INTRODUCTION

With the exponential growth in wireless communications, both for civilian and military usage, comes the need to cost effectively and quickly modify a radio device to meet new uses and to operate on different frequencies. In fact, the ideal would be a single physical radio that implements all existing communication signals and protocols (one radio that does it all) as well as one that adapts to new conditions to effectively use as many communications resources as possible, while still meeting applicable Federal Communications Commission (FCC) and World Radio Conference (WRC) requirements. Software defined radio (SDR) technology brings these benefits of flexibility and cost efficiency to end users. One established definition of Software Defined Radio is:

"Radio in which some or all of the physical layer functions are software defined"

A SDR typically includes a collection of hardware and software technologies where some or all of the radio’s operating functions are implemented through modifiable software or firmware running on devices including field programmable gate arrays (FPGA), digital signal processors (DSP), general purpose processors (GPP), programmable System on Chip (SoC) and other application specific programmable processors. This allows new wireless features and capabilities to be added to existing SDR systems without requiring new hardware.

There are several other terms which have slightly different meanings, but which we subsume under the category of software defined radio. Adaptive radio is radio in which communications systems have a means of monitoring their own performance and modifying their operating parameters to improve this performance. This capability enables greater degrees of freedom in adaptation to changing conditions and to achieve higher levels of performance and better robustness for a communications link. Cognitive radio is radio in which their communication systems are aware of both their internal state and the communications environment. Therefore they can make decisions about their operation by mapping that information against predefined objectives such as the most efficient usage of available frequency spectrum. Intelligent radio is cognitive radio that is capable of artificial intelligence, i.e. it is capable of automatically improving operation in the face of changes in environment and end user desires. All these concepts are typically implemented using SDR technology.

Software defined radios typically use a set of fixed modulation waveforms for a fixed set of channel bandwidths and choose among them to meet their operational goals. For example, when a set of non-contiguous channels become available, the SDR chooses one signal from this fixed set for each empty channel that matches that particular channel’s conditions (maximum energy, bandwidth, noise, channel characteristics, etc.). There are several drawbacks to this method. First, the fixed modulation waveforms do not exactly match each channel and so must be chosen conservatively so that all channel conditions are met, es-
pecially conditions that define the maximum signal leakage into adjacent channels. This mismatch leads to lower performance. Second, coding is typically done only within each channel, rather than across channels. This leads to worse performance for certain channels when they are affected by worsening channel impairment over time. Third, each channel must do its own synchronization (symbol, bit and frame, for example) because each is handled independently. This increases the complexity of the receiver due to the requirement of separate independent real-time parallel signal processing.

This paper describes a means of operation of typical software-defined radio systems through the on-demand custom design of modulation waveforms for communication on nearly any available spectrum under an unlimited set of operational conditions. In particular, this paper describes a method to do on-demand design of a communications waveform that spans the available spectrum within the digital bandwidth of the software-defined radio (SDR). This waveform creates a single modulation function across the (likely) non-contiguous available spectrum with near-optimal synchronization and correlation properties and with near-optimal bandwidth efficiency. Because of its designed structure, this modulation waveform can increase the bit-error rate performance and/or range of almost any existing wide-band SDR, while increasing robust performance under varying channel conditions. In addition, an SDR designed using the new hardware architecture described in this paper has a simpler structure than a traditional set of parallel digital modems, thus simplifying its design.

Many applications of our approach are possible for both the military and commercial communications. In the commercial arena, possible application areas include:

- Better spectrum usage for public channels such as ISM
- Operation within White Space (IEEE 802.22), such as within newly available UHF channels (formerly allocated to television)
- Adhoc wireless system and emergency service providers
- Mobile satellite and wireless systems such as new air-to-ground Connexion-type services
- Increasing manufacturing network capacity while operating heritage communications networks

The book [7] provides a good introduction to software defined radio concepts. The books [8] and [9] provide a good thorough theoretical framework for analyzing modulation signals of all types, while [10] gives a more thorough understanding of the system level communication design considerations.

II. Top Level Description

Figure 1 shows a standard software defined radio (SDR) system while Figure 2 shows the corresponding on-demand designed waveform architecture introduced in this paper. The three main differences are the replacement of a fixed set of modulation waveforms with a processing unit that solves an optimization problem to design a new modulation waveform particular to the measured channel conditions and required data rates, together with modified baseband processing using these new modulation waveforms.

Implementation of on-demand waveform design with SDR requires four items:

1. A method of mapping communications channel conditions into a mathematical description that can be used via standard optimization processes to generate the desired waveforms.
2. A numerically controlled waveform generator (NCWG) to implement the designed waveform in hardware or firmware.
3. A transmitter and receiver architecture that implements on-demand designed waveform communication using the NCWG.
4. On-demand waveform SDR operation implemented within the larger software defined network operation.

These four items are described in sections III-VI. Specifically, section III covers the mathematical description of the design process for modulation waveform generation; section IV covers the numerically controlled waveform generator; and section V describes the on-demand designed waveform transmitter and receiver architecture, amplifying what was shown in Figures 1 and 2 above. Finally, section VI describes the operation of an SDR that uses on-demand waveform generation as described previously.

The approach described was tested on several design problems and the results are described in section VII. Finally, this architecture was implemented using the Universal Software Radio Peripheral (USRP) platform (GNU radio hardware) from National Instruments/Ettus Research at Boeing’s 7-107 laboratory complex. This implementation and testing are described in section VIII.

III. On-demand Modulation Waveform Design

On-demand waveform design requires a method to map available channel conditions to a practical mathematical optimization problem. To map these channel conditions and, more generally, to map the problem of designing a modulation waveform to a mathematical optimization problem requires some preliminaries.

A. Delay estimation Cramer-Rao lower bound

Carrier, clock, bit, symbol or basically any other synchronization in a communications receiver requires measuring delay. In particular, part of the synchronization task is dependent on the modulation waveform rather than the data superimposed on this waveform. This is true in particular of symbol and bit synchronization. Thus the modulation waveform must have the property that it is robust to delay estimation in the presence of noise.

The Cramer-Rao lower bound (CRLB) is a lower bound on the estimation error or the error variance. It is given in its simplest
Figure 1: Standard SDR digital transmitter and receiver architecture.

Figure 2: On-demand designed waveform SDR digital transmitter and receiver architecture.
and most direct form for the delay estimation (hence synchronization) problem below. The problem is given an original transmitted signal \( s(t) \), to estimate \( \tau_0 \) given the received signal \( x(t) \) over time \( [0, T] \) where

\[
x(t) = s(t - \tau_0) + w(t), \quad 0 \leq t \leq T
\]

and \( w(t) \) is Gaussian noise. In its sampled form for software defined radio, the problem is to estimate \( \tau_0 \) given \( x[n] = x(n\Delta) \), \( n = 0, 1, ..., N - 1 \) where

\[
x[n] = s[n\Delta] - \tau_0] + w[n].
\]

The CRLB (see [1] for more information) for the variance of the estimate of \( \tau_0 \) given the noise variance \( \sigma^2 \) is

\[
\text{var}(\hat{\tau}_0) \geq \frac{\sigma^2}{\sum_{n=0}^{N-1} \left( \frac{\partial^2 s(t)}{\partial t^2} \bigg|_{t=n\Delta} \right)^2}, \tag{1}
\]

Note the dependence on the derivative of the original signal. The denominator can be closely approximated by the sum of the differences squared of the original signal

\[
\sum (s((n+1)\Delta) - s(n\Delta))^2.
\]

Thus by increasing the sum of differences squared, the lower bound on the variance of the delay estimate can be reduced, which in general will cause the estimators of this delay to be better. This is captured in the optimization criteria below by maximizing the \( L_2 \) norm of the approximate derivative of the signal, or equivalently minimizing its negative.

### B. WAVEFORM DESIGN CRITERIA AND OPTIMIZATION MODEL

The design of a modulation waveform must try to simultaneously optimize several things while meeting a number of criteria. A number of important ones are listed below, although this list is by no means exhaustive:

1. The waveform must be easy to synchronize between transmitter and receiver.
2. The waveform must meet energy or power restrictions required for communication signals required by regulatory rules and present conditions on the available channels.
3. The waveform must be designed to be robust in the presence of multi-path and/or interference.
4. The waveform must be bandwidth efficient; that is it must use all available channel capacity efficiently.
5. The waveform must meet spectral or frequency criteria required by regulatory rules and present conditions for communication signals on the available channels.
6. The waveforms should be easy to modulate; hence should involve linear modulation of a basis set of orthogonal waveforms.
7. The symbol error rate performance of such a designed waveform must be easy to estimate using conventional approaches.

These conditions can be captured in an optimization problem defined below. Let \( K \) be the number of orthogonal waveforms desired and \( N \) be the number of samples desired for each such waveform. Let \( \{s'_i\}, i = 0, 1, 2, ..., N - 1, j = 0, 1, 2, ..., K - 1 \) be these samples, i.e. the unknowns in the optimization problem below. To meet criteria 1 requires waveforms with good synchronization properties, i.e. because of the previous equation in the delay estimation results, it requires maximizing the \( L_2 \) norm of the derivative or equivalently minimizing the negative of the sum of the differences squared as given below. This must be applied to all the orthogonal waveforms as shown:

\[
\min_{s=(s'_i)} \sum_{j=0}^{K-1} \sum_{i=0}^{N-1} |s'_{i+1} - s'_i|^2.
\]

To meet the basic total maximum energy requirements (part of criteria 2) may be stated as

\[
\sum_{i} (s'_i)^2 \leq \text{total\_energy\_constraint}, \quad j = 0, 1, 2, ..., K - 1
\]

for each orthogonal waveform (and if time is factored in, this can also be phrased as a power constraint). This can be further refined to specify constraints in specific frequency bands as below (here \( \mathcal{F}() \) denotes the Fourier transform):

\[
|\mathcal{F}(s'_j)(F_m)|^2 \leq \text{band\_energy\_constraint}, \quad m = 0, 1, ..., M_F - 1
\]

for each such constraint on each of the \( M_F \) frequency ranges in \( \{F_m\} \). This is also the type of constraint used to satisfy criteria 4 by constraining energy to avoid certain frequency bands and therefore by implication to be forced to occupy only the complementary frequency bands. To meet the requirement that these \( K \) waveforms be orthogonal (or nearly orthogonal) and use linear modulation as in criteria 6 and 7 requires specifying this condition in a series of constraints:

\[
\sum_{i} s'_{ji}s'_i = 0
\]

for each pair \( \{j, j'\} \mid j, j' \in \{0, 1, 2, ..., K - 1\}, j < j' \) but it is usually better to instead specify a small value to which they are not allowed to exceed such as

\[
-\varepsilon < \sum_{i} s'_{ji}s'_i < \varepsilon \tag{2}
\]

or equivalently

\[
-2\varepsilon < 2 \times \text{total\_energy\_constraint} - \sum_{i} (s'_i - s'_i')^2 < 2\varepsilon.
\]

With \( \varepsilon \) small enough, the designed waveforms are orthogonal for all practical purposes.
To be robust in the presence of known multi-path or interference as in criteria 3 requires defining waveforms that avoid the multi-path or interference. If \( \{G_m\} \) is a set of frequency ranges with interference that must be avoided, a similar set of \( M_G \) band constraints could be formed as

\[
|\mathcal{F}(\{s^j_i\})(G_m)|^2 \leq \text{band\_energy\_constraint}, \quad m = 0, 1, \ldots, M_G - 1.
\]

By contrast, if the multi-path is estimated by doing channel equalization, then the inverse of the equalization filter, i.e. the channel filter function (written in finite impulse response form) defines a linear function on the time samples and the space orthogonal to the null space of this linear function defines a linearly defined region where the multi-path has the least affect. This can be written in the form below:

\[
(s^j_i) \times e_m = 0, \quad m = 0, 1, 2, \ldots, M_e - 1
\]

where \( \{e_m\} \) is a basis for the null space of dimension \( M_e \). Again, these are often more practically implemented using an inequality similar to Equation 2.

To meet criteria 5 using frequency constraints across multiple symbols (for example if operating without a separate pulse shaping filter), the constraints could be written as shown in Equation 3, where \( m_0 s, m_1 s, \ldots, m_{k-1} s \) denotes the time samples across \( k \) chosen symbols for each waveform \( s \in s^i \), and \( \{H_m\} \) are the frequency ranges involved. Here the set \( \{m_0^i, m_1^i, \ldots, m_{k-1}^i\} \) for each \( i \) is a chosen set of \( I \) sequences of modulation symbols of length \( k \) and there are \( I \) such constraints. Thus by choosing sets of representative symbol sequences and constraining the energy, out of band energy can be controlled in the design across multiple symbols.

The modulation symbols \( \{m_i^i\} \) could be binary symbols \( \{\pm 1\} \) if desired (and used for a one dimensional signal \( K = 1 \)) or from the QPSK set \( \{\pm 1, \pm i\} \) for a two dimensional signal for example. It is a good idea to start and end every base modulation waveform with zero to ensure continuity across the symbol boundary and this can be easily included into the above problem definition. Higher order modulation can be included easily as well. For example, if multiple base waveforms are desired, they can be easily accommodated without any increase in problem size by imitating the relationship between sine and cosine.

It is also possible to imitate sine and cosine more closely if desired, and this does have an impact on the numerically controlled waveform generator described in a later section. Not only are sine and cosine orthogonal, but they are also delayed versions of each other since \( \sin(s + \pi/2) = \cos(s) \). So an additional orthogonality constraint with \( n \) samples would look like:

\[
\sum_i s_{(i+P) \mod n} s_i = 0
\]

where \( P \) is the assumed delay in samples between the signal and its orthogonal delayed version. So for example, \( P = n/4 \) could be chosen to imitate sine and cosine, since they are the same function delayed by one quarter of their period or \( \pi/2 \).

C. Methods of Optimization

There are a number of standard approaches to the optimization step required as part of the design of the modulation waveform. The following will be a summary of two options that are practical to implement on embedded hardware. The first method is to use a constrained minimization of a non-linear multi-variable function. The constraints can be both linear and non-linear with both equality and inequality constraints as given here:

\[
\min_x f(x)
\]

subject to

\[
G_i(x) = 0, \quad i = 1, \ldots, m_e,
\]

\[
G_i(x) \leq 0, \quad i = m_e + 1, \ldots, m,
\]

where \( x \) is a vector of length \( n \), \( f(x) \) is the objective function which returns a scalar, and the vector function \( G_i(x) \) returns a length \( m \) vector corresponding to the equality and inequality constraints. The typical approach is to transform the problem into an easier sub-problem that can then be solved and used as the basis of an iterative process. In earlier approaches, the original constrained problem was translated to an unconstrained problem by using a penalty function for constraints that are near or beyond the constraint boundary. Then the constrained problem was solved by using a sequence of parametrized unconstrained optimizations, which in the limit converge to the solution for the constrained problem. These methods have been replaced by methods that solve the Karush-Kuhn-Tucker (KKT) equations. These equations are explored in [?] and [?] for optimization applications. The KKT equations are necessary conditions for optimality for a constrained optimization problem. If the problem is a so-called convex programming problem, that is, \( f(x) \) and \( G_i(x), i = 1, \ldots, m \), are convex functions, then the KKT equations are both necessary and sufficient for a global solution point. The solution of the KKT equations forms the basis of many nonlinear optimization algorithms. These algorithms attempt to compute the Lagrange multipliers in the KKT equations directly, often using a quasi-Newton updating procedure and are commonly referred to as Sequential Quadratic Programming (SQP) methods. Solution times for designed waveform problems can be done in time ranging from fractions of a second to several seconds depending on the problem details.

The second method explicitly uses quadratic programming and applies to a more specific subset of waveform design problems. In particular, it uses the concept of second order cone programs (SOCP). In a standard SOCP, a linear function is minimized over the intersection of an affine set and the product of second-order (quadratic) cones. SOCPs are nonlinear convex problems that include linear and (convex) quadratic programs as special cases. Several efficient primal-dual interior-point solution methods for SOCP have been developed recently and they are used in a variety of engineering design problems including filter design, antenna array weight design, truss design, and optimization in robotics. The second-order cone program (SOCP) is defined as:
|3|([m_0 s, m_1 s, ..., m_{l-1} s]) (H_m)|^2 \leq \text{multiple_symbol_band_constraint}, \ i = 0, 1, ..., I, \ m = 0, 1, ..., M_H \tag{3}

Multiple symbol band constraint.

\[
\min f^T x \\
\text{subject to} \\
|Ax + b_i| < c_i^T x + d_i, \ i = 1, \ldots, N,
\]

where \(x \in \mathbb{R}^n\) is the optimization variable. The paper [3] shows that the general convex quadratically constrained quadratic program (QCQP) can be solved by recasting it as an SOCP. The QCQP is defined as:

\[
\min x^T P_0 x + 2q_0^T x + r_0 \\
\text{subject to} \\
x^T P_i x + 2q_i^T x + r_i \leq 0, \ i = 1 ... p,
\]

where \(P_0, P_1, ..., P_p \in \mathbb{R}^{n \times n}\) are symmetric and positive semi-definite, \(q_0, q_1, ..., q_p \in \mathbb{R}^n\) and \(r_0, r_1, ..., r_p \in \mathbb{R}\). Since our norm constraints are clearly symmetric and positive semi-definite, we can often recast and solve the waveform design problems in this form. Solution times for designed waveform problems using SOCP methods can also be done in fractions of a second if properly formulated and are typically much quicker than using more general constrained optimization methods for a given problem size. The software [?] SeDuMi provides a convenient means for using SOCP methods from Matlab. For more general types of optimization software, the Wikipedia page [?] is a good starting point.

Our results in this paper have used FMINCON from Matlab (see [?]) which solves minimization problems subject to linear and non-linear constraints together with bounds. The particular algorithm within FMINCON used was the active set algorithm which solves the KKT equations. When used in the context of an embedded SDR radio system, it is usually best to limit the maximum number of iterations or optimization time to a portion of the total time allowed for switching to new waveform operation. Essentially this is determined by both practical hardware constraints as well as how fast RF environmental conditions are changing.

**IV. NUMERICALLY CONTROLLED WAVEFORM GENERATOR**

In order to use the baseband modulation waveforms designed and described in the next section in either a typical receiver or transmitter architecture (see Figures 7 and 8), a means must be created to take the baseband time samples of the waveform(s) and create intermediate frequency (IF) sampled waveforms at the proper IF frequency within the digital front-end of the transmitter (and correspondingly within the digital back-end of the receiver). For example, a set of baseband waveforms might be designed with a 1MHz sample rate for baseband frequency bands lying within 0-500KHz for use in the frequency range 49MHz-49.5MHz within the 50MHz digital front-end of the transmitter and then the entire 50MHz might be up-converted using the RF mixer to the RF frequency range of 1-1.05GHz for actual transmission. This is shown pictorially in Figure 3. The purpose of the numerically controlled waveform generator is to up-convert the baseband time sampled waveform digitally to the desired IF frequency just as a numerically controlled oscillator takes a fixed sine table and translates the sinewave to the desired frequency. The baseband and IF view are shown for a particular waveform during our testing and implementation in Figures 15 and 22.

For background information, we first describe a direct digital synthesizer (DDS). A DDS can be implemented from a precision reference clock, an address counter, a programmable read only memory (PROM), and a D/A converter as in Figure 4. The digital amplitude information that corresponds to a complete cycle of a sinewave is stored in the PROM in the SINE LOOKUP block. The address counter steps through and accesses each of the PROM’s memory locations and the contents (the equivalent sine amplitude words) are then presented to a high-speed D/A converter (DAC) which actually produces the analog “tone” or sine wave. However, this DDS lacks tuning flexibility since the output frequency can only be changed by changing the frequency of the reference clock or by reprogramming the PROM; neither of these options would allow high-speed continuous frequency adjustment as is required in an SDR.

If a phase accumulator function is added, this architecture becomes a numerically-controlled oscillator as shown in Figure 5. This Figure shows an N-bit variable-modulus counter and phase register before the sine look-up table, as a replacement for the address counter. The carry function allows this function to wrap as a new sinewave cycle is generated. Common DDS and NCO architectures exploit the symmetrical nature of a sinewave and utilize mapping logic to synthesize a complete sinewave cycle from ¼ cycle of data from the phase accumulator. An additional DAC
can be included to provide the cosine output from the phase-to-amplitude converter table which enables providing both I and Q outputs precisely matched in frequency, quadrature phase, and amplitude. Also, an inverse SINC block inserted before the DAC compensates for the $\sin(x)/x$ response of the quantized DAC output and therefore provides a constant amplitude output over the frequency range of the NCO. Finally, frequency/phase registers can be added to provide digital mixing to support specific types of modulation such as frequency-shift keying (FSK).

A numerically controlled waveform generator works in a similar manner as an NCO, but has a period that is controlled by the repetitive period of the designed modulation waveform (of length $N$ in the previous discussion). This is shown in Figure 6 below. By contrast with the sine wave, there is not necessarily a $\frac{1}{4}$ cycle symmetry in the designed waveform (unless this were also designed as part of the optimization), so an entire period must be stored. It works as follows:

- A tuning word is used to specify the value of the frequency translation desired. Thus in the previous example, a tuning word corresponding to a 49MHz translation would be specified to take the baseband waveform up to 49-49.5MHz from 0-500KHz.

- The tuning word is stored in a register and is used to accumulate the phase accumulator which is used to address the waveform lookup tables. A carry word is used to wrap at the end of the waveform period.

- The samples from the lookup tables are (optionally) interpolated for use by the modulation mixer where each symbol is multiplied by the waveform before summing and sending to the RF mixer in the transmitter. There is a corresponding similar usage of the NCWG in the receiver.

Note that the interpolation filter can be added as an option either to increase the sample rate beyond that of the optimization problem or to decrease the number of stored samples required in the look-up tables. This interpolation filter must be designed to reflect the constraints in the optimization in section C. For example, if $\{F_m\}$ is a set of disjoint passband frequency ranges that are desired for operation by the designed waveform, then a filter designed with these passbands and (of course) the complementary stopbands could be used for the interpolation filter with regard to this constraint. Note that these optional symbol interpolation filters are done on a per symbol basis and they do not necessarily obviate the need for a pulse shaping or transmitter interference filter that covers multiple symbols.

V. TRANSMITTER AND RECEIVER ARCHITECTURE

A simplified standard digital communications architecture for a transmitter is shown in Figure 7. Transmitters can vary widely in their particular architectures, so this is only meant to illustrate the changes necessary to incorporate a designed waveform into a transmitter architecture. The first five blocks describe the digital processing required to generate the waveform that is ultimately up-converted and broadcast. A receiver must reverse this process. The terms “cosine-like” and “sine-like” are short-hand notations for the typical linear modulation functions formed by multiplying I/Q symbols against cosine and sine functions generated by a numerically controlled oscillator driven at a particular center frequency.

The designed waveform transmitter architecture shown in Figure 8 uses the NCWG as described previously to generate the set of $K$ waveforms that were designed for the available channels. This set of waveforms is mixed with symbols generated from $K$ incoming bit streams and added to produce the transmitted baseband signal. An optional pulse shaping filter can be applied as well to mitigate intersymbol interference and meet adjacent leakage requirements. Note that the method of distributing $K$ channels of input bits (instead of the single stream in Figure 7) is system specific and is not further defined in this paper.

A simplified standard digital communications receiver architecture is shown in Figure 9 and the designed waveform receiver architecture is shown in Figure 10. Receivers can vary widely in their particular architectures (even more than transmitter architectures), so this is only meant to illustrate the changes necessary to incorporate a designed waveform into a typical receiver architecture. Note that the designed waveform architecture can combine the $K$ separate orthogonal symbol channels and operate on symbol vectors to reduce complexity of the receiver.

Of course, unlike the standard fixed modulation function, the receiver of the on-demand designed waveform must be informed about the waveform design so that it can receive the waveform correctly. This communication can be done either directly through a control channel with samples of the waveform, or indirectly through solving the same optimization design task using
Figure 7: Simplified standard digital transmitter architecture.

Figure 8: On-demand designed waveform digital transmitter architecture.

Figure 9: Simplified standard digital receiver architecture.

Figure 10: On-demand designed waveform digital receiver architecture.
the same input as the transmitter. Thus, this waveform sharing must be part of overall on-demand network operation described in the next section.

VI. ON-DEMAND MODULATION WAVEFORM OPERATION

Operation of an SDR system which uses on-demand designed modulation waveforms requires additional steps when used in a network setting. These are summarized below in a series of seven example steps and Figure 11 describe this process in a block diagram format:

1. SDR A senses which frequency bands (typically continuous segments of the radio spectrum) are available for use by SDR A and its network participants (those other SDR radios SDR B, SDR C, etc. that require communication with SDR A).
   • This sensing typically uses energy measurements within a band, but could also be done through a control channel.
   • The judgment of availability of a channel or channels may be further informed by on-board or control channel information from databases that define wireless regulation rules defined by government, world body, or local fixed rules.
   • The above considerations are typical of SDR and are not addressed further in this paper.

2. SDR A and its other network participants come to agreement about the above spectrum usage and its precise definition.
   • This could be done through a control channel or through an existing communication channel that is available to all participants.
   • This could also be done through one or more of the newly available bands using a predetermined signal modulation and type used for such control functions. This probing signal could also be measured and used to inform the waveform design process.
   • These steps are also typical of SDR systems which use probing signals to make simple communication decisions such as which fixed set of waveforms to use or even not to use the band at all. In the case of on-demand waveform design, the probing signal may need to be modified to assist in the optimization and waveform design process.

3. SDR A maps the sensed available chosen frequency bands to a set of mathematically defined constraints.
   • These are typically sets of contiguous pieces of the radio spectrum, denoted by the set \( \{ F_i \} \).
   • Each such piece typically has a maximum amount of transmitted power spillover defined by both regulation and the placement of nearby signals. This may be defined as either a maximum energy at a particular frequency within each \( F_i \) or as a total energy across this band or some combination of both.

4. SDR A maps the measured environment to a set of mathematically defined constraints.
   • The measured environment can be used to identify interfering signals (such as intentional or unintentional jamming tones for example). These interfering signals should not be present and would interfere with the available bands identified in step 1.
   • The measured environment can identify higher levels of noise which would negatively impact the symbol error rate.

5. SDR A solves an optimization design problem and produces the modulation waveforms and symbols to be used for network communication.

6. SDR A coordinates with other network nodes (SDR B, C, ... ) for network setup.
   • This coordination requires exchanging all relevant information about the network communication among all the participants.
   • An existing SDR must communicate many parameters, including the selection of modulation used in each band with its particular parameters, higher level packet framing and protocols and even high level network definitions which control how the newly created SDR network should operate.
   • Many of these items may be apriori known because of the nature of the participants and their knowledge of their fellow participants and their shared purpose, but anything that is dependent on the particulars of the environment at the point of time that the network is set up must be shared.
   • The previous items are part of normal SDR operation and will not be further described.
   • The new additional coordination that must take place for on-demand designed waveforms is the sharing of the waveform and symbol sets used by communicating the results of the optimization process.
This new item is not part of existing SDR operation.

7. Each network participant transmits and receives using the designed modulation waveforms.

VII. DESIGNING MODULATION WAVEFORM EXAMPLES

The techniques described in section B were used to design particular modulation waveforms subject to certain frequency and orthogonality constraints. This section describes three examples to show the performance advantages of our approach.

First Example. The maximum energy was upper bounded by 1 and the optimization used only a three symbol sequence design problem with the symbol sequences $[1, 1, 1]$, $[1, -1, 1]$ and $[1, -1, -1]$. Then the spectrum of a five symbol random sequence was evaluated and is shown in Figure 12. Note the 40dB reduction of out of channel power, without any need for separate pulse shaping. The time domain picture of one symbol is shown in Figure 13. Note that the optimization used only the available frequency as shown in Figure 12 pictorially in red. Here the lower red line represents the frequency region that is being constrained to be as close to zero as possible (stop-band) and the high red line denotes unconstrained (passband) frequency regions. The results of the optimization pushed most of the signal to the higher frequency side of the unconstrained portion of the spectrum, but each pulse is also shaped so as to reduce frequency leakage, just as if it had a pulse shaping filter applied. Finally, the signal variance energy or difference squared energy was increased by over 3.7db from the original signal used to start the optimization process (a signal with constant signal energy in the passband). This can be interpreted a number of ways using the CRLB in equation 1:

- The signal can be synchronized as precisely as before with over twice the noise.
- The signal can use a synchronization sequence that is half as long as before.
- The signal can be synchronized twice as fast as before.

Of course, these must be interpreted more precisely in the context of an overall communications system to realize such gains.

Second Example. As a second example, suppose two orthogonal signals are desired (hence an orthogonality constraint is added) with an additional set of three triple symbol spectral conditions $[1, i, i]$, $[1, -i, i]$ and $[i, i, i]$ used for the frequency constraint design. Then the spectral properties are similar to Figure 12 but the base signal in the time domain is much different. The twin orthogonal signal base pair is shown in Figure 14 (one in blue and the other in green). It is 3db better for synchronization than the original constant frequency domain default signal.

Third Example. As a third example, consider the problem where two non-contiguous frequency bands are available and a single waveform is desired to modulate across these bands.\(^2\) Furthermore, suppose that uniform power constraints have been put

\(^2\)Note that this assumes the antenna and front-end receiver can operate across these two bands to produce contiguous IF data.
across these bands at every occupied frequency. The spectral results for one particular case is shown in Figure 15 with time domain waveform as shown in Figure 16. Again, the lower red line shows where the frequency content was minimized, while the higher red line shows where the frequency content is unconstrained (but under uniform power constraints). In this case (where it was demanded that the same amount of power be present in each passband), the optimization did not place all the power in the higher frequency band, but did divide it among the two. Furthermore, note that the energy in the lower passband was pushed to the lower frequencies. This may seem counterintuitive, but in fact, this results in more energy in the signal derivative.

The processing time required for the optimization step in the three examples is shown in Table 1. These times are based on the fmincon function within Matlab and the machine is a Dell 490, vintage 2007. No attempt was made to reduce the times through
manipulation of the default optimization tolerances, or to pick a more efficient implementation such as one that could be used on an embedded processor in a radio. These questions are planned to be addressed in future work. Critical to this work would be reduction of optimization times so that in a radio system, the optimization (which could proceed as a background task) would need to be completed in a timely manner so that its result would be relevant to the current environment, rather than a past environment when the optimization problem was started.

VIII. USRP Test Setup and Implementation

We chose a Universal Software Radio Peripheral, or USRP, as our SDR test platform. The USRP is designed to allow general purpose computers to function as high bandwidth software defined radios and this allows testing of SDR algorithms on real radio hardware. Thus, the computer (or host) paired with the USRP serves as a digital baseband and IF section of a radio communication system. All waveform-specific processing, like modulation and demodulation, is done on the host CPU and, using the UHD driver, all of the high-speed general purpose operations like digital up and down conversion (DDC and DUC), decimation and interpolation are done on the FPGA internal to the USRP as shown in Figure 18. The USRP N200 includes a Xilinx® Spartan® 3A-DSP 1800 FPGA, 100 MS/s dual ADC, 400 MS/s dual DAC and Gigabit Ethernet connectivity for both host programming of the USRP and streaming USRP-received data back to host processors. The WBX daughtercard contains a wide bandwidth transceiver for simultaneous loopback operation, 100 mW of output power and a noise figure of 5 dB; as well as 40 MHz of bandwidth capability within its frequency range of 50 MHz to 2.2 GHz. The host/N200 pair is able to continuously and simultaneously process transmit and receive IQ data at 25MHz for 16 bit samples or 50MHz for 8 bit samples. The detailed specifications from Ettus Research are as shown in Figure 17. A picture showing the WBX daughtercard with separate transmit and receive SMA cables as shown in Figure 19.

Our test setup consists of the N200/WBX with loopback cables and a 30db attenuator along with various lab test equipment. A diagram of our setup is shown in Figure 20. To prove out our con-

<table>
<thead>
<tr>
<th>Type</th>
<th>Value</th>
<th>Unit</th>
<th>Type</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC Input</td>
<td>55</td>
<td>mV</td>
<td>128</td>
<td>35/70</td>
<td>dBc</td>
</tr>
<tr>
<td>Current Consumption</td>
<td>1.2</td>
<td>A</td>
<td>Power Frequency</td>
<td>100 Hz</td>
<td>100 dBc/Hz</td>
</tr>
<tr>
<td>sp WBB (per channel)</td>
<td>2.3</td>
<td>A</td>
<td>100 kHz</td>
<td>100 kHz</td>
<td></td>
</tr>
<tr>
<td>Frequency Accuracy</td>
<td>2.5</td>
<td>ppm</td>
<td>5 MHz</td>
<td>5 MHz</td>
<td></td>
</tr>
<tr>
<td>GPSDO Reference</td>
<td>1.2</td>
<td>ppm</td>
<td>Weight</td>
<td>1.2 kg</td>
<td></td>
</tr>
</tbody>
</table>

Figure 17: N200 USRP Specifications (source: http://www.ettus.com)

Figure 18: N200 USRP from Ettus Research (source: http://www.ettus.com)

Figure 19: USRP internals (N200 model) with WBX daughtercard showing dual TX/RX SMA cables.
Table 1: Optimization processing times for examples 1-3.

<table>
<thead>
<tr>
<th>Example</th>
<th>Optimization Time (sec)</th>
<th>Optimization Iterations</th>
<th>Waveform length</th>
<th>TolFun</th>
<th>TolCon</th>
<th>TolX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13.5</td>
<td>517</td>
<td>256</td>
<td>1e-9</td>
<td>1e-9</td>
<td>1e-6</td>
</tr>
<tr>
<td>2</td>
<td>87</td>
<td>649</td>
<td>256</td>
<td>1e-9</td>
<td>1e-9</td>
<td>1e-6</td>
</tr>
<tr>
<td>3</td>
<td>1308</td>
<td>1604</td>
<td>256</td>
<td>1e-9</td>
<td>1e-9</td>
<td>1e-6</td>
</tr>
</tbody>
</table>

cepts, we wrote custom C++ code using the UHD driver library to implement a loopback configuration of the N200 that would load designed waveform files and run a loopback test of random symbol transmission and reception. The main sequence of tasks for loopback are listed below:

1. Create a USPR device, lock onboard clocks and set the transmit and receive sample rate
2. Set the transmit center frequency and the RF gain, set the IF filter bandwidth and the transmit antenna
3. Set the receive center frequency and the receive RF gain, set the receive IF filter bandwidth and the receive antenna
4. Read in designed symbol IQ waveform (formed by the Hilbert transform of the interpolated symbol as shown in Figure 16).
5. Create a transmit streamer and check Ref and LO Lock detect and reset USRP time to prepare for transmit/receive
6. Start transmit worker thread and receive IQ data and write to output file

We chose the split spectrum symbol defined by optimization in example 3 previously and simultaneously send its IQ samples to the USRP while receiving digitized IQ data from the WBX receiver. The settings used on the USRP/WBX were:

- Number of samples = 200000000 (creates a 0.8GB file, runs for 8 seconds)
- Transmit rate = 25MHz
- Receive rate = 25MHz
- Transmit/Receive RF frequency = 1GHz
- IQ data size = 16 bits

The RF output is shown in Figure 21 on a spectrum analyzer, confirming the split spectrum of a stream of designed symbols as shown in Figure 15. The spectrum of the output file (containing the received IF) is shown in Figure 22.

IX. SUMMARY

In summary, we have described the concept of on-demand designed waveform generation and an approach to implement this concept in both standard and specialized software defined radio architectures. Specifically:
1. An optimization method is described to design modulation waveforms for SDR applications which enhances, among other things, both synchronization and bandwidth usage compared to traditional modulation methods. In particular, our approach allows:

   (a) Longer range and lower power communications under comparable conditions.

   (b) Communications allowing more efficient spectrum usage.

   (c) Spreading of synchronization and coding functions across multiple channels and disjoint spectral regions.

2. A numerically controlled waveform generator is described which allows hardware/firmware to create a sampled version of the designed modulation waveform at any particular frequency for modulation by the symbol sequence of a transmitter or conversely by correlating in the receiver. This waveform can be further tailored to reduce storage requirements through judicious interpolation and various symmetries can be imposed within the optimization to reduce storage as well.

3. An on-demand architecture is described that allows creation and utilization of a designed modulation function within a typical software defined radio architecture. A method of operation is also described that would allow on-demand waveform design to function within an existing SDR system.

Future work is planned to address real-world performance metrics to compare on-demand waveform design with traditional modulation approaches in given SDR scenarios. This work could also compare competing approaches such as OFDM to design waveforms for specific tasks such as spanning non-contiguous transmission bands. We also plan on addressing the computational requirements and latency in more detail, especially for embedded applications such as mobile radios, as well as coordination among radios using the proposed on-demand protocol.

REFERENCES


