Technical Introduction to Spiral Modulation

Technical Report R-2017-1

Dr. Jerrold Prothero
jprothero@astrapi-corp.com

Washington, D.C.

Last Update: February 28, 2017

Version 1.0

Copyright © 2017 Astrapi Corporation

Technology described in this document is covered by issued and pending Astrapi patents.
Technical Introduction to Spiral Modulation
Technical Report R-2017-1

Table of Contents
Acknowledgement & Disclaimer .................................................................................. 3
Abstract ..................................................................................................................... 4
1. Introduction ........................................................................................................... 5
2. Mathematical Background .................................................................................... 7
3. Instantaneous Spectral Analysis ........................................................................... 12
   Overview .................................................................................................................. 12
   Technical Description ............................................................................................. 13
   Detailed Operations .................................................................................................. 16
      Determine Signal Polynomial ................................................................................ 16
      Project Polynomial onto Cairns Series Functions ................................................. 17
      Convert from Cairns Series Functions to Cairns Exponential Functions .............. 21
      Combine Frequency Information .......................................................................... 21
4. Waveform Bandwidth Compression ...................................................................... 27
5. Spiral Polynomial Division Multiplexing ............................................................... 28
6. Single Spiral Modulation ....................................................................................... 30
7. Conclusion .............................................................................................................. 31

Table of Tables
Table 1: Generalized Euler’s Term as a Function of m ............................................... 7
Table 2: Number of Functions at Each m-Level ......................................................... 8
Table 3: The Cairns Series Coefficients ..................................................................... 8
Table of Figures

Figure 1: First Eight Cairns Polynomials ................................................................. 11
Figure 1: Generation of Random Polynomial ......................................................... 16
Figure 2: Conversion to Taylor Polynomial ............................................................. 17
Figure 3: Zero Pad Taylor Polynomial ................................................................. 18
Figure 4: Normalization Coefficients ................................................................. 19
Figure 5: Generate Cairns Projection Matrix ......................................................... 20
Figure 6: Project Onto Cairns Space ................................................................. 20
Figure 7: Find Amplitude Values ........................................................................ 21
Figure 8: Find Row Index .................................................................................... 22
Figure 9: Sort Frequencies .................................................................................... 22
Figure 10: Combine Amplitude Pairs ................................................................. 23
Figure 11: Reconstruction of Time Domain .......................................................... 24
Figure 12: ISA Time Domain and Frequency Domain Plots ................................ 25
Figure 13: Comparison of ISA and FT Spectral Usage ......................................... 26
Acknowledgement & Disclaimer
This material is based in part upon work supported by the National Science Foundation under Grant No. 1621082. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.
Abstract
A unified introduction to spiral modulation is provided, including its underlying mathematics and range of applications. Traditional digital telecommunication is based conceptually on complex circles derived from Euler’s formula, leading to transmission using sinusoids with constant amplitude over each symbol time. Spectral information is analyzed using the Fourier Transform (FT), which averages frequency information over a time interval. Spiral modulation is instead based on a generalization of Euler’s formula, leading to a new technique called Instantaneous Spectral Analysis (ISA) for representing signals as polynomials and converting them into sinusoids with continuously-varying amplitude for transmission. ISA is also a time-frequency analysis tool that makes it possible to describe the spectrum at a particular point in time, not averaged over a time interval as with the FT. By for the first time fully exploiting the capabilities of a nonstationary spectrum, spiral modulation makes it possible to dramatically increase spectral efficiency, through breaking an assumption in the sampling theorem that the spectrum is stationary. Spiral modulation provides potential benefits that may include increasing data throughput, reducing occupied bandwidth, reducing signal power, reducing latency, and mitigating phase noise and coherent interference.
1. Introduction

This document is an introduction to spiral modulation intended for a technical audience with a strong background in telecommunications. For more detailed information, which may be proprietary, please contact Astrapi.

Current digital communication has its foundations in 18th century mathematics. Conceptually, signals are defined by altering the parameters of complex circles, as described by Euler’s formula. The transmission model is a sequence of constant-amplitude sinusoids on a per-symbol basis, derived from complex circles. Spectral analysis is based on the Fourier Transform (FT), which uses Euler’s formula to convert a real-valued time sequence into a set of sinusoids each with constant amplitude. Effectively, the FT averages frequency information over a time interval.

Spiral modulation is based on a 21st century generalization of Euler’s formula that describes spirals in the complex plane. When applied to digital communications, this leads to a new technique called “Instantaneous Spectral Analysis” (ISA) that provides a transmission model based on sinusoids with continuously-varying amplitude. ISA also provides a tool for analyzing spectral information at each instant in time, not averaged over a time interval as with the FT.

We will refer to a spectrum as “stationary” if its frequency information does not vary over time, or equivalently if it is correctly represented by sinusoids with constant amplitude. If this condition does not hold, the spectrum will be termed “nonstationary”.

The transition from circles to spirals is a conceptually simple change with remarkably profound implications for communications theory and practice. It has been said that “Custom is second nature often mistaken for first”, and in the same way characteristics of the mathematics we have applied to communications theory have often been accepted as inherent characteristics of communications itself.

Notably, the following statements often (although not universally) associated with classical communication theory are affected significantly by spiral modulation.

1. A real-valued time sequence has a unique spectral representation provided by the FT. The FT averages spectral information across a time interval, which implies that the spectrum is assumed to be stationary over that period. The well-established field of time-frequency analysis exists because this assumption is in practice at best a reasonable approximation. For spiral
modulation, stationarity is a very poor approximation and examining its spectral content using an FT is inappropriate.

2. **Spectral occupancy should be measured in terms of FT roll-off.** Since the FT is not an appropriate tool for analyzing nonstationary spectra, it is also not an appropriate tool for measuring spectral occupancy if stationarity cannot be assumed. A spectral occupancy measure based on Inter-Channel Interference (ICI) is independent of a stationarity assumption and more economically meaningful.

3. **Sampling at the Nyquist rate is sufficient to fully reconstruct a bandlimited signal.** The Nyquist rate is a consequence of the sampling theorem, which cannot be proven without assuming the spectrum to be stationary. The spectrum is never stationary for spiral modulation.

4. **The Shannon-Hartley law is the upper bound on channel capacity.** The proof of the Shannon-Hartley law requires the sampling theorem and therefore implicitly assumes that the spectrum is stationary.

5. **The brick wall filter in the frequency domain is equivalent to the sinc function in the time domain.** This equivalence is a consequence of the FT. It therefore does not apply to cases in which the spectrum is not at least approximately stationary.

6. **A bandlimiting filter is necessary to constrain spectral occupancy.** Spiral modulation transmitted using ISA is inherently characterized by a sharp upper bound on the frequencies into which it places power. It therefore may not require a filter to remove power above some frequency range, although filters may be desirable for other reasons.

The most interesting consequence of spiral modulation is that, by for the first time fully exploiting the capabilities of a nonstationary spectrum, it in principle provides a means to exceed the Shannon limit on spectral efficiency. No traditional signal modulation technique based on sinusoids with constant amplitude over each symbol time is capable of doing this.

Direct benefits of the dramatically higher spectral efficiency made possible by spiral modulation include higher data throughput, lower signal power requirements, and reduced spectral occupancy. Other potential benefits of spiral modulation include lower latency and improved resistance to coherent interference and phase noise.

The following sections introduce the mathematical background and main implementation paths for spiral modulation.
2. Mathematical Background

This section provides the core mathematical background necessary to understand spiral modulation.

The familiar Euler’s formula

\[ e^{it} = \cos(t) + i \cdot \sin(t) \]  

(2.1)

can be generalized by raising the imaginary constant \( i \) on the left side to fractional powers. The new term

\[ e^{ti(2^2-m)} \]

(2.2)

reduces to the standard Euler’s term in the special case \( m = 2 \).

We can build the following table for positive integer values of \( m \):

<table>
<thead>
<tr>
<th>( m )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>( et^{i(2^2-m)} )</td>
<td>( e^t )</td>
<td>( e^{-t} )</td>
<td>( et )</td>
<td>( et^{i(1/2)} )</td>
<td>( et^{i(1/4)} )</td>
<td>( et^{i(1/8)} )</td>
<td>...</td>
</tr>
</tbody>
</table>

The standard Euler’s formula (2.1) can be proved by expanding \( e^{it} \) as a Taylor series and grouping real and imaginary terms. The same procedure can also be used for the term in (2.2) to derive a generalization of Euler’s formula for integer \( m \geq 0 \):

\[ e^{ti(2^{2-m})} = \sum_{n=0}^{[2^{m-1}]-1} in2^{2-m} \psi_{m,n}(t) \]

(2.3)

Where

\[ \psi_{m,n}(t) = \sum_{q=0}^{\infty} (-1)^q [2^{1-m}]. \frac{t^q[2^{m-1}]+n}{(q \cdot [2^{m-1}]+n)!} \]

(2.4)

The \( \psi_{m,n}(t) \) are called the “Cairns series functions” or the “Cairns polynomials”.
Notice that $\psi_{2,0}(t)$ and $\psi_{2,1}(t)$ give us the Taylor series for the standard cosine and sine functions, respectively.

Each value of $m$ produces a “level” of functions $\psi_{m,n}(t)$; from the summation limits in Equation 2.3, it can be seen that each level has a total of $[2^{m-1}]$ functions. That is:

<table>
<thead>
<tr>
<th>Table 2: Number of Functions at Each m-Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level (m-value)</td>
</tr>
<tr>
<td>Number of functions</td>
</tr>
</tbody>
</table>

An important property of the $\psi_{m,n}(t)$ is the regular pattern of their coefficients, as shown in the below table.

<table>
<thead>
<tr>
<th>Table 3: The Cairns Series Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\psi_{0,0}(t) = e^t$</td>
</tr>
<tr>
<td>$\psi_{1,0}(t) = e^{-t}$</td>
</tr>
<tr>
<td>$\psi_{2,0}(t) = \cos(t)$</td>
</tr>
<tr>
<td>$\psi_{2,1}(t) = \sin(t)$</td>
</tr>
<tr>
<td>$\psi_{3,0}(t)$</td>
</tr>
<tr>
<td>$\psi_{3,1}(t)$</td>
</tr>
<tr>
<td>$\psi_{3,2}(t)$</td>
</tr>
<tr>
<td>$\psi_{3,3}(t)$</td>
</tr>
</tbody>
</table>

Here, the rows indicate the Taylor series coefficients for each $\psi_{m,n}(t)$. For instance, the row for $\psi_{2,0}(t) = \cos(t)$ tells us that

$$\cos(t) = 1 - \frac{t^2}{2!} + \frac{t^4}{4!} - \frac{t^6}{6!} + \cdots$$ (2.5)
From Table 4 it is easy to see, and not difficult to prove, that the $\psi_{m,n}(t)$ coefficients define a set of orthogonal vectors. More precisely, if $M$ is a positive integer, then the vectors formed from the first $2^M$ coefficients of the functions $\psi_{m,n}(t)$ for $0 \leq m \leq M$ constitute a set of orthogonal basis vectors for a $2^M$-dimensional space. These can be normalized to produce the orthonormal “$2^M$ Cairns basis vectors”.

The existence of the $2^M$ Cairns basis vectors implies that any Taylor polynomial $P$ of degree $k < 2^M$ can be orthogonally projected onto polynomials formed from the first $2^M$ terms of the Cairns series functions simply by taking the inner product of $P$’s coefficients with the $2^M$ Cairns basis vectors. The resulting coefficients for each Cairns basis function are referred to as the “projection coefficients”.

The first $2^M$ terms of the Cairns series functions $\psi_{m,n}(t)$ are only an approximation to the full infinite series expansion of the $\psi_{m,n}(t)$. However, the error in the approximation is $O(t^{(2^M)})$, with a reciprocal factorial coefficient, and therefore falls off very rapidly as $M$ increases. For high-degree polynomials, therefore, it is reasonable to speak of projecting onto the $\psi_{m,n}(t)$ by this procedure.

It is well-known that the cosine and sine functions of Euler’s formula can be represented not only by Taylor series but also by sums of complex exponentials.

Explicitly:

$$
\cos(t) = 1 - \frac{t^2}{2!} + \frac{t^4}{4!} - \frac{t^6}{6!} + \cdots = \frac{1}{2} (e^{it} + e^{-it})
$$

and

$$
\sin(t) = t - \frac{t^3}{3!} + \frac{t^5}{5!} - \frac{t^7}{7!} + \cdots = \frac{1}{2i} (e^{it} - e^{-it})
$$

This characteristic also holds for the generalized Euler’s formula. We can define

$$
E_{m,n}(t) = \frac{1}{[2^m-1]} \sum_{p=0}^{[2^{m-1}] - 1} i^{-n(2p+1)}2^{2-m}e^{t(i(2p+1)2^{2-m})}
$$

Where the $E_{m,n}(t)$ are called the “Cairns exponential functions”.

By expanding the right side of Equation 2.8 as a sum of Taylor polynomials and recursively cancelling terms, it can be shown that for all $m$ and $n$
Equation 2.9 tells us that once a polynomial has been projected onto the Cairns series functions, it can be immediately converted into a sum of complex exponentials. This is useful for converting a polynomial into sinusoids with continuously-varying amplitude, which is the basis for the ISA algorithm discussed below.

As mentioned above, the Cairns basis functions allow any Taylor polynomial (which includes any polynomial with terms having positive integer powers) to be projected onto the Cairns series functions. Conversely, any weighted sum of Cairns series functions corresponds uniquely to a particular Taylor polynomial. As described below, Spiral Polynomial Division Multiplexing (SPDM) makes use of this fact in one of its instantiations to compose and decompose sub-channels. In the most straightforward implementation, each sub-channel corresponds to one amplitude-modulated Cairns function $\psi_{m,n}(t)$.

Let us look at the geometry of the generalized Euler’s term $e^{t \cdot i(x^{2-m})}$. By using the identity

$$e^{\frac{i\pi}{2}} = i$$  \hspace{1cm} (2.10)

which is a special case of Euler’s formula, it follows that

$$e^{t \cdot i(x^{2-m})} = e^{t \cdot \cos(\pi 2^{1-m})} \cdot e^{i \cdot t \cdot \sin(\pi 2^{1-m})}$$  \hspace{1cm} (2.11)

On the right side of Equation 2.11 the first factor, $e^{t \cdot \cos(\pi 2^{1-m})}$, is a real-valued exponential; the second factor, $e^{i \cdot t \cdot \sin(\pi 2^{1-m})}$, describes a circle in the complex plane. Their product describes a spiral in the complex plane.

As shown above, $m = 0$ generates the standard rising exponential; $m = 1$ generates the standard falling exponential; and $m = 2$ generates the standard sine and cosine functions.

From Equation 2.11 we can see that as $m$ increases above $m = 2$, the rate of growth increases and the rate of rotation decreases. In the limit as $m \to \infty$, $e^{t \cdot i(x^{2-m})}$ converges back to $e^t$. 

$$E_{m,n}(t) = \psi_{m,n}(t)$$  \hspace{1cm} (2.9)
At every level of $m$, the $\psi_{m,n}(t)$ share the growth and frequency properties of their generating term $e^{t(2^m-1)}$. For instance, the Cairns functions with the fastest rotation (and slowest growth) occur at $m = 2$.

Another important observation is that for $m \geq 2$ Cairns functions with even $n$ are always symmetric around $t = 0$, and Cairns functions with odd $n$ are always anti-symmetric around $t = 0$. This generalizes a well-known feature of the cosine and sine functions.

The Cairns polynomials through $m = 3$ are plotted below for the interval $-3.1 \leq t \leq 3.1$.

![Cairns Polynomials Graph](image)

**Figure 1: First Eight Cairns Polynomials**
3. Instantaneous Spectral Analysis

Overview
ISA converts a sequence of \( N \) amplitude values (the “time domain”), or equivalently a polynomial of degree \( N - 1 \), into a sum of sinusoids with continuously-varying amplitude. This is in contrast to the FT, which converts the time domain into a sum of sinusoids with constant amplitude.

For a particular amplitude sequence, define \( B_F \) to be the range of sinusoidal frequencies occupied by the FT; let \( B_I \) be the range of sinusoids with non-zero power as generated by ISA.

Below are key differences between ISA and the FT.

1. Since the FT represents an amplitude sequence with a basis set of sinusoids having constant amplitude, it assumes an evaluation period over which the spectrum is stationary; that is, over which the power assigned to particular frequencies is constant. This assumes that the source of the amplitude sequence is Linear Time-Invariant (LTI). ISA does not require an LTI source.

2. The FT effectively averages spectral information over its evaluation period to produce constant sinusoidal amplitudes. ISA is capable of determining continuously-varying sinusoidal amplitudes at every instant in time (hence the name “Instantaneous Spectral Analysis”).

3. For the FT, the maximum rate at which independent amplitude values can be transmitted is equal to the Nyquist rate of \( f_N = 2B_F \). For the ISA, there is no inherent upper bound in terms of \( B_I \) on the rate at which independent amplitude values can be transmitted. ISA holds this advantage over the FT because Shannon’s proof of the sampling theorem, from which the Nyquist rate derives, assumes that the spectrum is stationary over the evaluation interval. ISA violates this assumption.

4. Stating the above point in a different way, using ISA it is possible to transmit a sequence of amplitude values in a given amount of time using a dramatically smaller range of frequencies with nonzero amplitude than is possible with the FT representation.

5. Effectively, for a given amplitude sequence time duration, the FT responds to a higher level of detail in the amplitude sequence by utilizing higher frequencies (increasing \( B_F \)). ISA responds by increasing the density of sinusoids within a constant \( B_I \).
We are used to referring to the output of the FT as the frequency domain, but really it is a frequency domain; it provides a possible representation of the time domain in terms of sinusoids with constant amplitude. ISA provides a different frequency domain representation in terms of sinusoids with continuously-varying amplitude. This raises the question of how we should think about occupied bandwidth: is the FT representation of occupied bandwidth “correct”, or is the ISA representation “correct”?

The answer is that it depends on how the signal was generated. If the source was LTI over the evaluation interval, then the FT representation is correct.

However, ISA provides a recipe for generating a non-LTI signal source that will dramatically reduce the necessary range of frequencies with nonzero amplitude necessary for transmission; for such a source, the FT representation is incorrect.

Using ISA, a sequence of amplitude values can be transmitted such that the product of the bandwidth $B_I$ (defined in terms of the range of frequencies containing non-zero power) times the signal duration time $T$ is always equal to one. Since a polynomial of degree $D$ corresponds to $D+1$ independent amplitude values, this makes it possible to exceed the Nyquist rate. This occurs because the proof of the sampling theorem, from which the Nyquist rate derives, assumes that the spectrum is stationary over the evaluation interval. ISA violates this assumption.

Practically, bandwidth measurement matters because it affects how closely channels can be packed together without Inter-Channel Interference (ICI). The real test of ISA bandwidth efficiency is the extent to which it allows for closer channel packing.

**Technical Description**

The key steps of ISA are as follows.

1. **Represent a signal as a polynomial.** This can occur either by fitting a polynomial to a sequence of amplitude values, or by selecting or generating the signal polynomial directly. This topic is covered in more detail in the sections below on Waveform Bandwidth Compression (WBC) and Spiral Polynomial Division Multiplexing (SPDM).

2. **Project the signal polynomial onto the Cairns series functions $\psi_{m,n}(t)$.** The projection occurs in the polynomial coefficient space.

3. **Convert from the Cairns series functions $\psi_{m,n}(t)$ to the Cairns exponential functions $E_{m,n}(t)$.** This is simply a labelling change as the two
representations are equivalent numerically; however, the Cairns exponential functions provide frequency information explicitly that is implicit in the Cairns series functions.

4. **Combine frequency information contained within the Cairns exponential functions.** Group terms with the same frequencies to provide a representation of the signal polynomial in terms of a sum of sinusoids with continuously time-varying amplitudes.

The mathematics for the second and third steps were provided in 2. Mathematical Background. A description of the fourth step, and MATLAB® software for all four steps, is provided here.

While spectral usage is not readily apparent from the $\psi_{m,n}(t)$ representation, it can be determined precisely, and on an instant-by-instant basis, from the equivalent $E_{m,n}(t)$. Essentially, projection onto $\psi_{m,n}(t)$ allows us to decompose a polynomial, and then representing it as $E_{m,n}(t)$ allows us to generate an equivalent set of sinusoids with continuously-varying amplitude.

Each $E_{m,n}(t)$ can be expressed as a sum of products in which each term is the product of a phase-adjusted real-valued exponential with a complex circle. The real-valued exponentials may be either rising or decaying, and may have different growth constants in the exponent. The complex circles may rotate in either direction, and with different frequencies.

By viewing the real-valued exponentials as continuously time-varying coefficients applied to sinusoids, we have an approach to defining instantaneous spectrum. At any particular time, the sum of the real-valued exponentials applied to complex circles of the same frequency defines the spectral usage at that particular frequency at that particular time.

In more detail, by using the Euler’s formula identity $e^{i\pi/2} = i$ it is possible to represent $E_{m,n}(t)$ as

$$E_{m,n}(t) = \frac{1}{[2^{m-1}]^2} \sum_{p=0}^{[2^{m-1}]^2-1} i^{-n(2p+1)2^{1-m}} e^{t \cdot \cos(\pi(2p+1)2^{1-m})} e^{i \cdot t \cdot \sin(\pi(2p+1)2^{1-m})}$$

In the above equation, the phase is determined by $i^{-n(2p+1)2^{2-m}}$, the amplitude by $e^{t \cdot \cos(\pi(2p+1)2^{1-m})}$, and the frequency by $e^{i \cdot t \cdot \sin(\pi(2p+1)2^{1-m})}$. 
The instantaneous spectral information can be found by summing the phase-weighted amplitude information associated with each frequency.

Note the following points:

- For \( m = 0 \) and \( m = 1 \) the frequency factor is equal to the constant 1
- For \( m \geq 2 \), no two distinct \( m \) levels will contain the same frequencies, since \( \sin(\pi(2p+1)2^{1-m}) \) depends on \( m \).
- The same frequency appears in \( E_{m,n}(t) \) for every \( n \) at level \( m \), since \( \sin(\pi(2p+1)2^{1-m}) \) does not depend on \( n \).
- Since both \( \sin(\pi(2p+1)2^{1-m}) \) and \( \cos(\pi(2p+1)2^{1-m}) \) can switch signs depending on the value of \( p \), it follows that for \( m \geq 2 \) each positive frequency will be matched by an equal negative frequency, and that for \( m \geq 3 \) each positive and negative frequency will appear with both a rising and falling exponential as its real-valued amplitude coefficient.

To find the instantaneous amplitude of each frequency, we have to algebraically assemble all terms that have the same frequency. Since (as noted above) the frequency factor does not depend on \( n \), the same frequency information will appear in all \( E_{m,n}(t) \) functions having the same \( m \). We therefore have to sum phase-adjusted terms across \( n \) values at the same \( m \) level.

Since a particular frequency is fully-determined by the combination of its \( m \) and \( p \) values, let us denote a given frequency by \( f_{m,p} \) and its amplitude at a particular time by \( a_{m,p}(t) \). Then we have

\[
a_{m,p}(t) = \frac{1}{[2^{m-1}]^{1-m}} \sum_{n=0}^{[2^{m-1}]-1} c_{m,n} i^{-n(2p+1)2^{2-m}} e^{t \cdot \cos(\pi(2p+1)2^{1-m})}
\]

This equation provides the instantaneous frequency amplitudes associated with the polynomial \( P \) at every distinct time \( t \) over its evaluation interval. As noted above, for \( m \geq 3 \) each frequency will appear twice, associated with each of a rising and decaying exponential amplitude. These paired amplitudes should be summed together.
A detail is that the $c_{m,n}$ were found by projecting Taylor coefficients onto the $\psi_{m,n}(t)$ normalized by their number of non-zero terms. However, $\alpha_{m,p}(t)$ reconstructs the signal polynomial from $E_{m,n}(t)$ terms that have not been normalized in this way. Since $E_{m,n}(t) = \psi_{m,n}(t)$, the same projection normalization factor as applied to the $\psi_{m,n}(t)$ must be applied for matched $m, n$ values; this was left out for readability but is performed in the below software.

**Detailed Operations**
The below subsections provide the detailed operations for the corresponding high-level steps given above.

**Determine Signal Polynomial**
As discussed above, a signal can be viewed as equivalent to a polynomial or sequence of polynomials. A polynomial for signal transmission can arise either from fitting a polynomial to a sequence of data-carrying amplitude values, or by generating the polynomial directly, as described in the WBC and SPDM sections below.

Assuming the signal is given to us as a sequence of real-valued amplitude values, standard techniques exist for fitting a polynomial to the sequence of signal amplitude values, and others are currently under investigation at Astrapi. ISA does not depend on which method is used. Usual considerations apply, such as whether to produce a high-order polynomial that exactly fits the data sequence, or a lower-order polynomial that is an adequate fit but perhaps better captures the underlying pattern. The only concern for ISA is that a polynomial be provided that represents the time domain amplitude sequence.

For illustrative purposes, the following figure provides MATLAB code to generate a polynomial of arbitrarily chosen $25^\text{th}$ degree, with random Taylor coefficients between -10 and 10.

```matlab
% Generate random Taylor poly of arbitrary positive integer degree
degree = 25; % Arbitrary polynomial degree
n_idx = degree:-1:0; % Powers of all poly terms
nfact_reciprocal = 1./factorial(n_idx);
Taylor_coefficients = 20*(rand(1,degree+1)-0.5); % Rand Taylor poly
taylor_poly = Taylor_coefficients .* nfact_reciprocal;
```

**Figure 2:** Generation of Random Polynomial
Project Polynomial onto Cairns Series Functions
This operation converts a polynomial $P$ representing the signal time-domain data into a weighted sum of Cairns series functions.

First, we convert $P$ into a Taylor polynomial by multiplying through by the factorial of each term’s power. In the below MATLAB code sample, a polynomial is represented as a row vector of coefficients called `poly_coefficients`, with the highest power on the left. The resulting `Taylor_coefficients` represents the same polynomial with the factorials implicit.

```matlab
% Example:
% The polynomial coefficients row vector [3/3! 2/2! 1 0]
% will be returned as [3 2 1 0]
[rows, len] = size(poly_coefficients);
Taylor_coefficients = zeros(rows, len);
for r=1:rows
    idx = 0:len-1;
    facts = factorial(idx);
    descending_facts = fliplr(facts);
    Taylor_coefficients(r,:) = poly_coefficients(r,:) .* descending_facts;
end
```

**Figure 3: Conversion to Taylor Polynomial**

The code in Figure 3: Conversion to Taylor Polynomial is capable of handling a matrix in which each row represents a distinct polynomial, although for current purposes a single row is sufficient.

To shorten the notation, we refer to “Taylor_coefficients” equivalently as “Taylor_vec” below.

Next, if necessary we pad “Taylor_vec” with minimum-value high-term coefficients so that the number of coefficients is a power of two, i.e. $2^M$ for positive integer $M$. This is because our polynomial coefficient projection is based on a table with a power-of-two number of rows and columns.
We next construct the Cairns projection matrix.

The following MATLAB code finds the normalization coefficient for each row of the Cairns projection matrix.

```matlab
% Find polynomial degree+1
[~,len] = size(Taylor_vec);

% Cairns functions increase by powers of two
M = nextpow2(len);

% If necessary, pad Taylor coefficient vector to be right length
if len ~= 2^M
    if len > 2^M
        error('Internal error');
    end
% Pad to right degree
    new_Taylor_vec = zeros(1,2^M);
    pad_by = 2^M - len;
    new_Taylor_vec(1,1:pad_by) = 1/realmax;
    new_Taylor_vec(1,pad_by+1:end) = Taylor_vec;
    Taylor_vec = new_Taylor_vec;
end
```

**Figure 4: Zero Pad Taylor Polynomial**

We next construct the Cairns projection matrix.
% Number of terms in Cairns function Taylor series approximation
len = 2^M;

% Normalization coefficient for each row
norm_vec = zeros(1, len);

% Find normalization coefficients
this_row = 1;
for m = 0:M                       % Over all levels
    for n = 0:ceil(2^(m-1))-1    % Over all functions at this level
        % Specify the non-zero elements for this row
        step_size   = ceil(2^(m-1)); % Dist between non-zeros
        num_non_zero = floor(len/step_size); % # of non-zero entries
        norm_vec(this_row) = 1/sqrt(num_non_zero); % Each row magnitude 1
        this_row = this_row +1;
    end
end

**Figure 5: Normalization Coefficients**

Given the normalization coefficients, the following code generates the Cairns projection matrix.
Given the Cairns projection matrix, our Taylor polynomial can be projected onto Cairns space by simple matrix multiplication. The following code produces a row vector in which each row is the coefficient for a Cairns series function.

```matlab
% Compute the projection matrix in ascending degree
proj_matrix = zeros(len);
this_row = 1;
for m = 0:M       % Over all levels
    for n = 0:ceil(2^(m-1))-1 % Over all functions at this level
        step_size = ceil(2^(m-1)); % Distance between non-zeros
        if m==0
            row_idx = 1;
        else
            row_idx = 2^(m-1) + n + 1; % +1 because rows start at 1, not 0
        end
        norm_by = norm_vec(1,row_idx);
        num_non_zero = floor(len/step_size); % Number of non-zero entries
        for q = 0:num_non_zero-1;
            proj_matrix(this_row, q*step_size + n + 1) = ... 
                (-1)^(n+1)*norm_by;
        end
    this_row = this_row +1;
end

% Convert to descending degree, for consistency with MATLAB convention
proj_matrix = flipud(proj_matrix);
```

**Figure 6: Generate Cairns Projection Matrix**

**Figure 7: Project Onto Cairns Space**
Convert from Cairns Series Functions to Cairns Exponential Functions
Because of the identity $\psi_{m,n}(t) = E_{m,n}(t)$, the conversion from Cairns series functions to Cairns exponential functions is automatic. We simply consider the projection coefficient for each $\psi_{m,n}(t)$ to be applied to the corresponding $E_{m,n}(t)$.

However, because each $\psi_{m,n}(t)$ was normalized as described above, the same normalization factors must be applied to the $E_{m,n}(t)$. In the below code, these normalization coefficients are held in the `proj_norm` row vector.

Combine Frequency Information
The below code shows how to combine amplitude information associated with each frequency by summing across $E_{m,n}(t)$ $n$-values. Note that the amplitudes are time dependent (specified by the evaluation time $t$).

```matlab
% Effectively, the below reconstructs the input from the E(m,n); but in % order to pull the frequency information out, we are summing across % n-values. The projection coefficients we are using (Cmn, below) come from % projecting onto normalized Psi(m,n) functions. So we need to normalize % the E(m,n) in the same way, using the projection normalization % coefficients.
proj_norm = compute_norm_coefficients(M);

this_freq_idx = 1;
for m=0:M % Over all m-levels
  max_n = ceil(2^(m-1))-1; % Cairns functions at this level
  max_p = ceil(2^(m-1))-1; % Frequencies at this m-level
  n_norm = 1/ceil(2^(m-1)); % m-level normalization factor
  for p=0:max_p % Over all frequencies
    freq_idx(1,this_freq_idx) = sin(pi*(2*p+1)*2^(1-m));
    sum_this_p = 0;
    for n=0:max_n % Over all functions at this m-level
      Cmn = proj_coeffs(1,mn_to_row_index(m,n)); % Our proj coeff
      phase = 1i^(-n*(2*p+1)*2^(2-m));
      amp = exp(t*cos(pi*(2*p+1)*2^(1-m)));
      sum_this_p = sum_this_p + Cmn*phase*amp;
    end
    idx = mn_to_row_index(m,0);
    freq_amp(1,this_freq_idx) = sum_this_p * n_norm * proj_norm(1,idx);
    this_freq_idx = this_freq_idx+1;
  end
end
```

**FIGURE 8: FIND AMPLITUDE VALUES**
The function `mn_to_row_index` converts from a pair of \( m, n \) values to the corresponding matrix index. It was given in-line in Figure 6: Generate Cairns Projection Matrix above, and is defined as follows.

```
if m==0
    row_index = 1;
else
    row_index = 2^(m-1) + n + 1;  % +1 because rows start at 1, not 0
end
```

**Figure 9: Find Row Index**

Frequency values across \( m \)-levels are interleaved, so next we sort.

```
% The frequency information overlaps across m-levels, so sort
[freq_idx, sort_idx] = sort(freq_idx);
freq_amp = freq_amp(sort_idx);
```

**Figure 10: Sort Frequencies**

We now add for each frequency the rising and decaying exponential amplitudes. In the process, the frequency vectors are shortened.
We now have the instantaneous frequency and amplitude information stored in `short_freq_idx` and `short_freq_amp` respectively.

The last step is to show how the instantaneous spectral information found above can be used to reconstruct the time domain at a particular time value $t$. 
The accuracy of the reconstruction increases with the degree of the polynomial representing the signal, since the projection onto Cairns space becomes more precise with longer polynomials. For instance, for a random 25\textsuperscript{th} degree polynomial maximum percentage reconstruction ratio errors are less than $10^{-10}$.

The left panel of the following figure shows the time domain for a random 15\textsuperscript{th} degree Taylor polynomial (solid red line) and its five positive-frequency ISA components traced over time (dashed red lines).

The right panel of the figure shows the ISA sinusoidal amplitudes for matched positive and negative frequencies at the arbitrarily-chosen time of $t = 0.20$. 

% Integrity check. The frequency information should be able to reconstruct % the value of the polynomial at the specified time. 

```matlab
poly_at_t = polyval(poly_vec,t);  % Time-domain value
if abs(poly_at_t) > 0.00000001     % Ratio error doesn't work if zero
    our_sum = 0;                % Reconstruct from freq domain
    for short_idx = 1:short_len
        amp = short_freq_amp(1,short_idx);
        freq = short_freq_idx(1,short_idx):
        our_sum = our_sum + amp*exp(1j*t*freq);
    end
    reconstruction_error = abs(our_sum - poly_at_t);
    percent_error = 100*reconstruction_error/abs(our_sum);
    if percent_error > 1
        poly_vec
        short_freq_idx
        short_freq_amp
        t
        poly_at_t
        our_sum
        reconstruction_error
        percent_error
        error('Reconstruction error greater than 1%');
    end
end
```

**Figure 12: Reconstruction of Time Domain**

The accuracy of the reconstruction increases with the degree of the polynomial representing the signal, since the projection onto Cairns space becomes more precise with longer polynomials. For instance, for a random 25\textsuperscript{th} degree polynomial maximum percentage reconstruction ratio errors are less than $10^{-10}$.

The left panel of the following figure shows the time domain for a random 15\textsuperscript{th} degree Taylor polynomial (solid red line) and its five positive-frequency ISA components traced over time (dashed red lines).

The right panel of the figure shows the ISA sinusoidal amplitudes for matched positive and negative frequencies at the arbitrarily-chosen time of $t = 0.20$. 

Figure 13: ISA Time Domain and Frequency Domain Plots

In this case, the maximum ratio percentage error in the ISA reconstruction of the polynomial across the entire evaluation interval is $75 \times 10^{-5}$; the mean percent reconstruction error is $7.3 \times 10^{-1}$. The accuracy of the ISA reconstruction improves as the polynomial degree increases, since longer Taylor series are used.

Notice that the simulated transmission time duration $T$ is 1 microsecond, and the range of frequencies $B$ in which ISA puts power is exactly 1 MHz. Since a random 15\textsuperscript{th} degree polynomial transmits 16 independent amplitude values, the number of amplitude values that can be transmitted in this way is $16BT$. This is 8 times higher than the $2BT$ limit provided by the sampling theorem on the assumption (which ISA breaks) that the spectrum is stationary over the transmission interval.

This difference is apparent from comparing the mean ISA spectral power with the FT of the amplitude sequence generated by the random 15\textsuperscript{th} degree polynomial, as displayed below.
**Figure 14: Comparison of ISA and FT Spectral Usage**

In this figure, as shown in the left panel all ISA power is placed within the 1MHz range. The FT in the right panel is on a frequency larger scale, showing only a roughly 35dB roll-off at 10MHz.
4. Waveform Bandwidth Compression

The ISA algorithm detailed above gives us the ability to transmit signals using sinusoids with continuously-varying amplitude, rather than the traditional sinusoids with constant amplitude per symbol time. Doing so potentially allows data throughput to be increased by transmitting independent amplitude values at a rate higher than the Nyquist rate. As discussed above, this is possible because the proof of the sampling theorem rests on an assumption that the spectrum is stationary, which ISA violates.

Waveform Bandwidth Compression (WBC) arises from asking the question: “What is the simplest way to obtain benefits from ISA while changing existing radio architecture as little as possible?” The WBC answer is to intervene in the encoder of traditional signal modulation methods such as Phase-Shift Keying (PSK) or Quadrature Amplitude Modulation (QAM).

Any digital signal modulation technology produces a sequence of data-carrying amplitude values that it intends to transmit using sinusoids. WBC views that sequence of amplitude values as defining a sequence of polynomials of some degree. Applying ISA then makes it possible express the data-carrying amplitude values into a much smaller range of frequencies with nonzero amplitude than would be necessary using traditional methods. This has the effect of compressing the waveform bandwidth. In principle, the receiver design is not affected by WBC because the expected sequence of amplitude values arrives, although expressed differently in terms of sinusoidal sums.

A rough analogy, for purposes of building intuition, is that the same shape Chinese parade dragon can be produced with more or less people underneath, depending on how the people are deployed. In a similar way, WBC can deliver the same sequence of amplitude values as traditional transmission using a smaller range of nonzero frequencies, by deploying the frequencies differently (exploiting continuous sinusoidal amplitude variation).

For further details on WBC, please contact Astrapi.
5. Spiral Polynomial Division Multiplexing

WBC, discussed above, provides a minimally-invasive approach to applying ISA to traditional radio architecture. Spiral Polynomial Division Multiplexing (SPDM) takes a different approach. SPDM arises from asking the question: “If we were going to design radio communications from scratch using ISA, what would we do?”

Since ISA potentially provides us for the first time with a bandwidth-efficient means to transmit arbitrary polynomials, it opens the possibility of designing a communication system to use a basis set of polynomials for its symbol waveforms.

The SPDM model is that data is encoded using amplitude modulation of a basis set of polynomials, and transmission occurs using ISA. This provides a very natural link between digital data and analog waveform transmission. Every polynomial of degree $D$ is equivalent to a sequence $D + 1$ independent amplitude values. The polynomial itself describes an analog waveform suitable for transmission using ISA; the value of the polynomial sampled at specified times can convey digital data.

Since general polynomials have much more waveform distinguishability than the sinusoids used by traditional signal modulation methods, they are more resistant to noise, and therefore may have much better Bit Error Rate (BER) performance as a function of $E_b/N_0$.

Within the SPDM broad umbrella are a number of topics.

1. **Choice of basis polynomials.** A straight-forward implementation of SPDM is to amplitude-modulate a basis set of Cairns polynomials, although other polynomial basis sets are also possible. For instance, to convey 8 bits per symbol (comparable to 256-QAM), one could support two amplitude levels for each of eight basis polynomials. An 8-bit sequence is then transmitted by producing a “transmission polynomial” corresponding to the weighted sum of the basis polynomials, and generating a sequence of amplitude values from the transmission polynomial. The task of the receiver, given a sequence of amplitude values, is to determine the polynomial that generated the sequence, and therefore the transmitted message.

2. **Signal detection algorithm.** The receiver can use standard minimum Euclidean distance signal detection, and/or other techniques arising from the properties of the polynomial basis set.
3. **Synchronization.** The use of polynomials provides new capabilities for synchronization, which are the subject of National Science Foundation (NSF)-funded research.

4. **Single or multi-user configurations.** Different communication sub-channels can optionally be assigned to subsets of the polynomial basis set that is combined into the transmission polynomial (hence the “Division Multiplexing” in SPDM).

5. **Coherent interference rejection.** To the extent that coherent interference can be described by a set of polynomials, linear algebra can potentially be applied to generate an SPDM polynomial basis set that is orthogonal to the coherent interference, thus mitigating its effect.

Potential benefits of SPDM include

1. **Much better BER vs. \( E_b/N_0 \) performance** than traditional signal modulation techniques such as QAM, due to the much greater symbol waveform distinguishability of general polynomials compared to simple sinusoidal modulations.

2. **Bandwidth-efficient transmission** using ISA.

3. **Less vulnerability to phase noise** since SPDM does not explicitly encode information into phase.

4. **Less vulnerability to coherent interference** due to new SPDM-specific mitigation techniques involving rotation of the basis polynomial set.

5. **Lower signal power requirements** since the higher spectral efficiency of SPDM can be used to reduce signal power, for the same communication performance.

6. **Lower latency** since SPDM spectral efficiency can be used to reduce Forward Error Correction (FEC) overhead.

For further details, please contact Astrapi.
6. Single Spiral Modulation

The primary focus of this document has been on ISA-based techniques that exploit sums of complex spirals. However, a more basic approach is also possible, which modulates the generalized Euler’s term in much the same way that traditional signal modulation uses the standard Euler’s term, but with an extra parameter related to the continuous complex amplitude variation of a spiral.

ISA as described in previous sections makes use of an integer-valued $m$ parameter. However, we can replace this with a real-valued $g$ parameter, and we can treat $g$ as a parameter that can be modulated.

$$e^{ti(2^2-g)}$$

Fixing $g = 2$ would give us the standard Euler’s formula and support standard techniques such as amplitude and phase modulation. Allowing $g$ to take on values other than 2 opens a means to modulate spiral waveform shape. Other ways of modulating spiral waveform shape are also possible, notably controlling the spiral complex amplitude using splines to smooth discontinuities.

It is generally useful to return the complex amplitude to its original value at the end of each symbol time, to minimize discontinuities. This involves a “ramping up” and “ramping down” period (or the reverse).

When compared to a comparable PSK implementation, one advantage of single spiral modulation can be much faster side-lobe roll-off as measured using a traditional FT, since intra-symbol discontinuities can be reduced by assigning a low complex amplitude to the inter-symbol boundary.

There are other possible benefits of single-spiral modulation, including the ability to send reference signal information that also carries traffic.

For further details, please contact Astrapi.
7. Conclusion

Astrapi has pioneered a new form of communication based on complex spirals, rather than the traditional complex circles. Based on a solid mathematical foundation and for the first time fully exploiting the capabilities of a nonstationary spectrum, spiral modulation potentially opens the door to a range of benefits centered around higher spectral efficiency. The ability to exploit a much wider symbol waveform design space based on polynomials also makes available new techniques for coherent interference rejection and other benefits.