A Function Modulation Method for Digital Communications

Subhendu Das  
CCSI, West Hills, California  
subhendu.das@ccsi-ca.com

Avtar Singh  
San Jose State University, California  
asing@email.sjsu.edu

Nirode Mohanty  
CCSI, West Hills, California  
nirode.mohanty@ccsi-ca.com

Abstract

We present a novel modulation method for digital communication systems. The method uses orthogonal and/or non-orthogonal analytic functions as symbols. It is shown that, in the case of orthogonal functions, to transmit $M$ bits using one symbol, the new receiver needs to search over only $M$ functions as opposed to $2^M$ symbols. A numerical method for generating the analytic symbols, called Constrained Gram Schmidt method, is presented. Experimental results demonstrating the proof of concept of the approach are also discussed.

1. Introduction

The foundation of the existing digital communication schemes is based on modulating the three parameters, amplitude $A$, frequency $f$, and phase $\phi$ of the sinusoidal function:

$$s(t) = A \sin(2\pi ft + \phi)$$

Most of the existing methods vary, in discrete steps, one or more of the above three parameters to represent the digital data. These methods are known as Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK), and Phase Shift Keying (PSK).

In this paper we present a modulation method for digital communication that does not use sinusoidal functions, does not use any one of the keying methods mentioned, and does not use any kind of discrete variations during the symbol intervals as well as at the inter-symbol interfaces. We show that the entire symbol stream remains analytic.

In the proposed method we modulate the complete function $s(t)$ or the $\sin$ function itself. Since we are modulating the functional structure of the expression $s(t)$, we call it a function modulation (fm) method. Also, since our method does not use any discrete changes in the waveform or the function representing the symbol we call it an analog approach. In this paper we will use the term, lower case fm, to refer to the new modulation method.

Our design approach is based on the concept of Software Radio (SWR), where we use batch data-in and batch data-out processing method as opposed to more normal sample-in and sample-out type real time method. This SWR approach allows us to see the past, the present, and therefore the entire history of the data simultaneously, and to help extract information more effectively at the receiver. In this SWR method we do not need to use Voltage Controlled Oscillators (VCO) or Phase Lock Loops (PLL) etc. The design can be implemented entirely on a standard off-the-shelf Digital Signal Processor (DSP).

To avoid proliferation of variable names we have reused the same name many times just like in software. We believe that to a careful reader the context will help to distinguish them.

2. fm Transmitter

Fig. 1 describes the design of a transmitter of the function modulation (fm) method for digital communication system. The left-hand-side vertical box represents a four-bit data, as an element of data space, to be transmitted using one symbol $s(t)$. In Fig.1 we have assumed, the number of bits, $M$, to be transmitted is four, for example, without any loss of generality.

---

Figure 1. fm Modulation Method

1-4244-0697-8/07/00 ©2007 IEEE.
Let \( d = \{ d_i, \ i=1..M, \ d_i \in \{0,1\} \} \), be a column vector, and represent a data element in the data space. Here \( d_i \) represents the i-th bit of \( d \) with 0 or 1 as its value. Let \( G(t) = \{ g_i(t), \ i=1..M, \ t \in [0,T] \} \), also a column vector, represent a set of analytic and independent functions defined over the symbol interval \([0,T]\). We assign the i-th function \( g_i(t) \) to the i-th bit location. In Fig. 1 the arrows from the bit locations to the bit-function boxes define the one-to-one assignments. These functions are referred to as bit-functions. The set \( G(t) \) defines the bit-function space.

A set of functions \( G(t) \) is called dependent if there exists constants \( c_i \), not all zero, such that

\[
g_1(t)c_1 + g_2(t)c_2 + \ldots + g_M(t)c_M = 0
\]

for all \( t \) in \([0,T]\). If not then it is independent [1, pp 177-181]. The above expression is a linear combination of functions. Here the coefficients \( \{ c_i, \ i=1..M \} \) are all real numbers.

A function is analytic if it has finite valued derivatives. Analytic functions are band limited [2, p75]. An analytic function does not have any discrete jumps; it is a smooth and continuous function. In this paper the terms analog and analytic functions are used interchangeably.

The \( M \) bit-functions are combined inside the algorithm box to produce one symbol function \( s(t) \). The collection of all symbols is called the symbol space. The Fig.1 shows how we have introduced the concept of a bit-function space in between the data space and the symbol space.

The algorithm is selected in such a way, so as to produce a symbol that is also an analytic function. For every bit pattern in the data space the algorithm produces one unique symbol in the symbol space using only \( M \) bit-functions from the bit-function space. The algorithm is an one-to-one and onto transformation, from the data space to the symbol space, ensuring that for every symbol it produces, there exists one unique bit pattern.

This idea of representing every bit by an analog function and then combining them by an algorithm to produce another analog function helps us to map the complete digital data as an analog symbol. Using this method we can transmit digital data as analog signals.

In general the algorithm in Fig. 1 can be represented by the following expression:

\[
s(t) = A[ G(t), d ]
\]

In (1) \( A \) is an arbitrary algorithm, operator, or transformation, which can be algebraic or dynamic, as well as, linear or non-linear. The operator \( A \) is a mapping from the product of the bit-function space and the data space to the symbol space. A simple example of the operator \( A \) can be given by the following (2):

\[
s(t) = d'G(t) = g_1(t)d_1 + g_2(t)d_2 + \ldots + g_M(t)d_M
\]

The notation \( ' \) indicates the transpose of the column vector \( d \). (2) is not a linear combination in the strict sense. The coefficients in (2) are not real numbers; they are only 0-1 integers. We will call this algorithm as a 0-1 addition algorithm. In this paper (2) will define the fm transmitter.

When all bits are zeros, expression (2) does not produce a meaningful symbol. To circumvent this problem we use a special, predefined analytic function, consistent with the technology presented here, to represent the symbol when all bits are zeros. The receiver will first test for the presence or absence of this special function, to detect the transmitted data corresponding to all zero bits, before using the standard algorithm discussed later.

We observe that this 0-1 addition process has a very interesting mathematical consequence. This approach can be used to invert the algebraic addition process. That is, if we add two numbers say 2 and 3 to produce 5, then given 5 we can find out which two numbers were added. Representing the integers 0 through 9 by 10 different continuous functions we can do this. Zero can be represented by zero function. This concept can generate interesting consequences in mathematical theories.

The fm transmitter concept may appear similar to the OFDM concept [3]. But there are some major differences. OFDM uses one or more of the three basic modulation methods (ASK, FSK, and PSK) from the existing technology, fm does not. OFDM configures the spectrum bandwidth into disjoint regions but fm does not. In fm every bit-function spans the entire bandwidth of the channel. OFDM uses only harmonically related sinusoidal orthogonal functions; fm does not need to use any orthogonal function. fm can use non-sinusoidal orthogonal functions as well as non-orthogonal functions, OFDM cannot. However, under certain restrictive conditions OFDM is considered as a special case of fm technique. If we select only sinusoidal and harmonically related orthogonal bit-functions, use only amplitude modulation with zero or full signal variation, and use 0-1 addition algorithm, then fm is same as OFDM.

This fm approach also has some similarities with the Quadrature Chaos Shift Keying (QCSK) scheme [4]. There are some fundamental differences though. The QCSK is a shift keying method therefore it generates discrete
variations in the symbol; both inside the symbol intervals as well as at the symbol boundaries. The chaos functions are high bandwidth functions, the objective of the fm approach is to use band limited or analytic functions to represent the symbols. The QCSK transmits a reference signal over a portion of the symbol. The QCSK uses orthogonal functions [4]. We have demonstrated in this paper that the fm approach can use non-orthogonal functions as well as orthogonal functions. The similarity is that the QCSK uses non-sinusoidal functions just like the fm method.

3. fm Receiver

At the receiver we will receive the symbol function \( s(t) \) as shown in Fig. 1, corrupted by the noise and/or the non-linearities of the communication channel. Our objective at the receiver will be to find out which bit-functions from the bit-function space \( G(t) \) were used to generate the received symbol. That is, we have to decompose the received symbol into the component bit-functions. The presence or absence of a bit-function in the received symbol will indicate 1 or 0 value, respectively, for the bit at the corresponding bit location.

A set of bit-functions \( G(t) \) is orthogonal if the following holds:

\[
\int_0^T g_j(t)g_i(t) \, dt = 0 \quad \text{for} \quad i \neq j, \quad i, j = 1..M
\]

Observe that we have defined orthogonality over finite time interval \([0,T]\).

All sinusoidal functions are orthogonal over infinite time interval. Only harmonically related sinusoidal functions are orthogonal over a finite time interval. It is easy to verify, using the above relation, that the two sinusoidal functions with frequencies 1000 Hz and 1100 Hz are not orthogonal over the period 1/1000 seconds or 1/1100 seconds. It is also well known [5] that there are infinitely many, band-limited, non-sinusoidal, orthogonal functions over a finite time interval. However there are only a finitely many band limited sinusoidal orthogonal functions over a given finite time interval.

The 0-1 addition formula gives the expression for the received symbol, \( r(t) \), as shown below. The following (3) is a derived from (2).

\[
r(t) = g_1(t)x_1 + g_2(t)x_2 + \ldots + g_M(t)x_M + w(t)
\]

Here, \( w(t) \) is an Additive White Gaussian Noise (AWGN) process, \( x_i \) are the bit values, unknown to the receiver but known to the transmitter and are equal to \( d_i \). If we assume that the bit-functions in \( G(t) \) are orthogonal then we can find \( x_i \) using the following (4) simple relation:

\[
x_i = \int_0^T r(t)g_i(t) \, dt + w_i
\]

In above \( w_i \) is the projection of \( w(t) \) over \( g_i(t) \). We can set the bit values \( d_i \) using the following relation:

\[
d_i = 1 \text{ whenever } x_i > 0 \text{ otherwise } d_i = 0.
\]

A fm receiver design that uses orthogonal functions is shown in Fig. 2.

![Figure 2. fm Receiver with orthogonal functions](image)

Fig. 2 is identical to a standard figure in many communication textbooks. However it has a few significant differences also. Notice that it has only \( M \) parallel paths as opposed to \( 2^M \) parallel paths found [6, p135] in the existing methods. The output from each correlator is the bit-value of the corresponding bit location, which is not the case in conventional methods. In conventional methods only one of the boxes produces an output indicating a symbol match in the corresponding path. You can also find a similar figure in textbooks that uses orthogonal functions [2, p569] and that has \( M \) parallel paths. In that figure the output of each path is a real number. In Fig.2 the outputs are only 0 or 1 integers representing the bit values.

Fig. 2 and equation (4) show that for the fm method, based on orthogonal functions, we need to search over only \( M \) functions in the bit-function space as opposed to \( 2^M \) symbols in the symbol space. Thus the orthogonal fm method can significantly reduce the complexity of the receiver design. The introduction of the bit-function space in between the symbol space and the data space helps us to achieve this interesting result.

The receiver design for the non-orthogonal case is quite complicated and involved. The design is not unique
also. In the remaining part of this section we describe one numerical algorithm for the design of a fm receiver that uses 0-1 addition algorithm and non-orthogonal set G(t). That is, all the bit-functions, \{g_i(t)\}, are now analytic and independent only. The set G(t) is known to both the receiver and the transmitter. Given the information in (3) our problem at the receiver is, again, to solve (3) for 0-1 integer values for the unknown variables \{x_i\}.

Note that, in this formulation, the problem (3) is not really a classical 0-1 Integer Programming Problem (IPP). There is no optimization function associated with the equality expression (3). There is a random noise variable in (3) which is also not found in the standard IPP. Also the coefficients in (3) are not real numbers but functions of time.

There are various methods available in the scientific and engineering literature for solving the above receiver problem. In this paper we discuss only one of them. We convert the problem (3) to a least square solution problem by sampling, at fixed intervals, all the signals K times over the symbol period \([0,T]\), where \(K\) is an integer greater than or equal to \(M\). Thus (3) can be expressed by the following set of simultaneous linear equations:

\[
[r(t_1)] = \begin{bmatrix} g_1(\tau_1) & g_2(\tau_1) & \cdots & g_M(\tau_1) \\ g_1(\tau_2) & g_2(\tau_2) & \cdots & g_M(\tau_2) \\ \vdots & \vdots & \ddots & \vdots \\ g_1(\tau_K) & g_2(\tau_K) & \cdots & g_M(\tau_K) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{bmatrix} + \begin{bmatrix} w(\tau_1) \\ w(\tau_2) \\ \vdots \\ w(\tau_K) \end{bmatrix}
\]

Here, \(\{\tau_1, \tau_2, ... \tau_K\}\) are equally spaced sample points inside the time interval \([0,T]\). We will assume \(K\) is larger than \(M\) giving us more equations than the number of unknown variables \(M\). Using the matrix notation the problem defined by (3) can then be rewritten as in (5).

\[
r = Ax + w
\]

Since \(G(t)\) is a set of functions with analytical expressions they can be sampled any number of times. The length of the vector \(r\) can also be increased by interpolation between real samples obtained from Analog to Digital Converters (ADC). Thus the number of samples need not depend on the sample rate of the ADC or on other electronics in the receiver. This is one of the advantages of using the software radio approach. We can always get more number of equations than the number of unknowns giving us a better least square solution for (5).

In (5) \(r\) and \(w\) are \(K\)-column vectors with components consisting of \(K\) samples of the functions \(r(t)\) and the AWGN process \(w(t)\), respectively. \(A\) is a \(K\times M\) rectangular matrix with elements defined by (6), and \(x\) is the unknown 0-1 column vector \([x_1, x_2, ... x_M]^T\) taken from \([x_i]\).

\[
a_{ij} = g_j(t_i), i = 1..K, j = 1..M, t_i \in [0,T], K > M \tag{6}
\]

Since the functions in the set \(\{g_i(0)\}\) are independent the matrix \(A\) with elements defined by (6) is a full rank matrix. Therefore \(A' A\) is non-singular and the real valued solution of (5) can be expressed by (7) using the pseudo inverse \(P\) of \(A\) [7].

\[
x = P r = (A' A)^{-1} A' r, \quad \text{where} \quad P = (A' A)^{-1} A' \tag{7}
\]

The bit values \(\{d_i\}\) can then be obtained by the following decision logic (8):

\[
d_i = 1, \quad \text{when} \quad x_i \geq \beta \quad \text{and} \quad d_i = 0 \quad \text{otherwise} \tag{8}
\]

The threshold value \(\beta\) is a given constant representing the channel characteristics.

The pseudo inverse gives a least square error solution of the simultaneous linear equations (5). It essentially curve fits the received symbol function \(r(t)\) using the bit-functions of the set \(G(t)\). Note that the matrix \(P\) is constant for a given fm system and can be precomputed and stored in memory. The accuracy of the fm system can be controlled by controlling the number of samples, \(K\), for each function.

4. fm Characteristics

The Power Spectral Density (PSD) of \(s(t)\) and the Bit Error Rate (BER) expressions have been derived. The BER has been derived for the case of orthogonal functions only. The PSD expression is valid for both orthogonal and non-orthogonal functions as bit-functions.

The complete symbol stream \(s(t)\) for the 0-1 addition algorithm and for all time \(t\) can be given by (9).

\[
s(t) = \frac{1}{M} \sum_{k=-\infty}^{\infty} \sum_{m=1}^{M} d_m(k) g_m(t - kT) \tag{9}
\]

The factor \(M\) was introduced to normalize the amplitude of the symbol. Since each bit-function is normalized, the addition of \(M\) bit-functions requires renormalization by \(M\). Note that \(d_m\) is the bit value, 0 or 1, and is not the multilevel value of the data element even though we are considering \(M\)-bit data. Also note that there
is no constant pulse shaping signal associated with the symbol stream. The bit-functions replaced them.

It is interesting to observe that the structure of the above mathematical expression is similar to that of the standard OFDM [3, p5-8].

We denote the bit correlation function by

\[ R_{mn}(p,q) = \mathbb{E}[d_m(p)d_n^*(q)] \]

and its two sided Fourier Transform (FT) by

\[ C_{mn}(v,w) = \sum_{p=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} R_{mn}(p,q)e^{-j2\pi pvT}e^{-j2\pi qwT} \]

Using the above two definitions and substituting \( G(f) \) as the FT of \( g(t) \), the PSD can be represented by (10):

\[
S(f) = \frac{1}{M^2} \sum_{m=1}^{M} \sum_{n=1}^{M} C_{mn}(-f,f)G_m^*(f)G_n(f) \quad (10)
\]

Although (9) is similar to OFDM, the final expression (10) for \( S \) is quite different. Here we have reused the symbol \( G \) with different meaning and we hope that the context helps to prevent confusions, if any. The absence of periodic pulses in (9) removed all discrete terms from the expression (10).

As an example of the PSD result, we use the following sinusoidal functions, used in Multiple Phase Shift Keying (MPSK) systems, as the bit-functions in \( G(t) \):

\[
g_{m}(t) = A \cos(2\pi f_c t + \theta_m) \quad -T/2 \leq t \leq T/2
\]

Substituting the above sine functions in (10) and performing some algebraic simplifications, we can arrive at the following PSD expression for this case of fm scheme.

\[
S_{fm}(f) = \left( \frac{AT}{2} \right)^2 \left( \frac{M \sin \frac{\pi f MT}{M}}{\pi f T} \right)^2
\]

Comparing the above PSD expression with that of the MPSK [8] system, given below,

\[
S_{MPSK}(f) = A^2 T \left( \frac{\sin \pi f T}{\pi f T} \right)^2
\]

we see that the fm spectrum requirement is narrower as \( M \) increases. The PSD graphs of the two systems are plotted in Fig. 3. The graphs for fm system are plotted for four different bit data length \( M \).

We summarize the BER result here for the orthogonal case only. Consider one of the parallel paths of the orthogonal fm detection method presented in Fig.2. The transmitted symbol \( s(t) \) in fm modulation scheme using orthogonal functions can be derived from Fig.1 and expressed by (11):

\[
s(t) = \sum_{i=1}^{M} x_i \sqrt{E_i} g_i(t) \quad (11)
\]

In expression (11) we assume that \( x_i = -1 \) if the bit is zero and +1 if the bit is one, \( \{g_i\} \) is a set of \( M \) orthonormal functions and \( \{E_i\} \) is the energy of the orthonormal signal for bit \( i \). Using expression (11) we can show that the BER probability is given by:

\[
P_{B_E} = Q\left( \sqrt{\frac{2E_i}{N_0}} \right)
\]

The above result shows that the BER for orthogonal fm receiver is same as that of the BPSK scheme.

5. Symbol Generation

In this section we describe a numerical method for computing the bit-functions used in fm system. The bit-functions, \( G(t) = \{g_i(t), i=1,..M\} \), can be generated using various methods. We express the bit-functions as a linear combination of sinusoidal functions as shown in (12). Expressions (12) will ensure that the bit-functions in \( G(t) \) are analytic within the symbol interval \([0,T]\).

\[
g_i(t) = \sum_{j=1}^{M} c_{ij} \sin(\omega_{ij} t + \phi_{ij}) \quad t \in [0,T]
\]

In equation (12) \( \{\omega_{ij}\} \) and \( \{\phi_{ij}\} \) are some arbitrary and convenient choices for generating the functions. \( \{\omega_{ij}\} \) must be within the channel bandwidth making sure that \( G(t) \) is a band limited set. Each one of the functions in \( \{g_i(t)\} \) are
Since each set of sine functions for each \( g_i(t) \) are independent, their linear combinations are also independent making \( G(t) \) an independent set. The value of \( M_c \) will depend on the number of constraints defined below by (13-17).

The constant coefficients \( \{c_{ij}\} \) of the linear combination in (12) are selected to satisfy a series of constraints defined below. We only mention some of the constraints that appeared to be necessary for the proper operation of the fm system as defined in this paper. Different communication channel may require different set of constraints. However the general concept presented here still covers many possibilities.

To synchronize the symbols at the receiver we make the symbols start and end at zero value (13). A sufficient condition for that is to make the bit-functions behave the same way. Thus we consider the following constraints on \( \{g_i(t)\} \):

\[
g_i(0) = 0, \quad g_i(T) = 0, \quad i = 1..M
\]  

To make the symbols join smoothly we want to impose the following derivative constraints (14) at the two ends of the bit-functions.

\[
\left. \frac{d}{dt} g_i(t) \right|_{t=0} = \alpha, \quad \left. \frac{d}{dt} g_i(t) \right|_{t=T} = \alpha, \quad i = 1..M
\]  

In (14) \( \alpha \) is any real number. In this paper we will set \( \alpha=0 \) merely for convenience. It is clear that if we want further smoothness at the symbol interfaces we can force the higher order derivatives to similar constraints. The constraints (13) and (14) will ensure that the entire symbol stream given by (9) is analytic. They also prevent any kind of discrete variations at the inter-symbol interfaces.

In many situations it may be necessary to avoid biasing the communication channel by a Direct Current (DC) voltage. To implement that requirement we set the integrals of all bit-functions to zero, as given by (15):

\[
\int_0^T g_i(t) dt = 0, \quad i = 1..M
\]  

To be able to detect the symbols properly at the receiver we may need to make the symbols pass through some predefined points (16). We call them way points. This property may also help to synchronize the symbols properly.

\[
g_i(t_k) = a_k, \quad t_k \in [0,T], \quad k = 1..K_i, \quad i = 1..M
\]  

Here, \( \{K_i\} \) denotes the number of way points for the i-th bit-function and \( \{a_k\} \) are some known choices.

If we want to generate orthogonal functions as bit-functions then we include the following constraints (17):

\[
\int_0^T g_i(t) g_j(t) dt = 0, \quad i, j = 1..M, \quad i \neq j
\]  

We call the method described above as the Constrained Gram Schmidt (CGS) method. The CGS method allows us to generate orthogonal functions with specified characteristics.

Summarizing, the method for generating the bit-functions is to substitute the expression for the bit-function (12) into all the constraints (13-17) defined above. This substitution will produce several linear equations for the unknown constants \( \{c_{ij}\} \). This set of simultaneous equations can then be solved for the constants. These constant coefficients will then be substituted back in (12) to get the analytical expression for each bit-function. Note
that for each bit-function we have to solve a different set of equations. The above process generates $G(t)$ as an independent set of analytic functions with specified bandwidth.

We have used the constraints (13-16) to generate four non-orthogonal bit-functions. That is, we did not use the orthogonality constraints defined by (17). These bit-functions are shown in Figures 4 and 5.

6. Proof of Concept

At this stage of development our main concern was to learn the characteristics of the received signal. That is, how well it retains its transmitted properties. Prior to the experimental verification we were using simulation with AWGN based corruption of the transmitted signal. Once we had the hardware, we have tried on the real environment, and our approach has completely changed. This section briefly describes this hardware board, experimental setup, and the results.

![Figure 6. Hardware board functional block](image)

The block diagram of this off-the-shelf hardware board that we found is described by Fig.6. These are two identical TMS320C5402 DSP boards [9] from Texas Instruments (TI). The board has a telephone line interface with a Data Access Arrangements (DAA) Integrated Circuit (IC). This DAA takes care of the voltage conditions and protection of the telephone line. The TLC320AD50 is an IC codec and contains both an Analog to Digital Converter (ADC) and a Digital to Analog Converter (DAC).

The board has a printer parallel port for interfacing with the computer. Via this printer port we control the boards using the TI Code Composer Studio (CCS) [10] software development tools. These boards allow us to perform the real time experiment on the Plain Old Telephone System (POTS) network.

For this experiment we operate the codec for both receiver and transmitter at 16 kHz sample rate with 16-bit data resolutions. The symbol duration used was one millisecond. Only the transmission and reception of symbols via the telephone line are performed in real time using the hardware boards. The symbol creation and the symbol analysis functions are performed off line using Mathematica.

The experimental setup is shown in Fig.7. We assemble everything, both transmitter and receiver, in one laboratory room with two telephone sockets having two different telephone numbers. The two DSP boards are connected to the telephone lines and to two computers to control them.

The bit-functions shown in Figures 4 and 5 are used to generate the symbol corresponding to the bit pattern 1011. This transmitted symbol is shown in Fig. 8. We transmit this symbol several times using our laboratory setup and capture the received signals, shown in Fig.9, at the receiver end. All received signals looked almost exactly the same.

![Figure 8. Transmitted fm symbol](image)

It is interesting to observe that in Fig.9 we did not encounter any standard AWGN in the system. The telephone line noise is significantly lower than what we initially anticipated in this experiment. In figures 8 and 9
the horizontal axes are one millisecond long. However the nonlinear signal distortion, as seen in Fig.9, which has two positive peaks instead of three, is quite severe and unexpected. We are not sure how and why this distortion happened. It may be due to the 8 KHz sampling rate of POTS and 16 KHz sampling rate of our boards. We are expecting that a very high quality hardware board may help to minimize or eliminate the distortions.

To synchronize the received signal we perform linear interpolation and up sampling; then remove few points from both ends to make the received symbol start and end very close to the time axis. Finally we use (7) to solve for the least square curve fitting problem for $K=17$. The result is the following values for the unknown variables $\{x_i\}$:

$$\{1.8653, 0.45037, 1.03662, 0.851587\}$$

A threshold value of 0.5 for $\beta$ in (8) gives the bit values for correct transmitted data, 1011.

A threshold value of 0.5 for $\beta$ in (8) gives the bit values for correct transmitted data, 1011.

Even though we encounter severe non-linear distortions we are still able to recover the bits correctly using only 17 samples. This shows that the curve fitting method, using the bit-functions along with the 0-1 addition algorithm, is indeed very robust. Note that it is also the availability of the entire data history that played a very important role in extracting the information.

7. Conclusion

The existing digital communication theory is based on discrete concepts. This concept is then translated to discrete variations of sinusoidal functions. In this paper we presented a new approach, where we represent digital data using analytic non-sinusoidal functions that does not use any kind of discrete concepts.

We also use DSP and software approaches for the design of our receiver. The batch-data-in and batch-data-out buffers, with very fast processors can help in the information extraction process. The batch data approach will allow us to use many mathematical theories that were never used before in communication engineering.

The digital data representation concept discussed here can be extended to data compression techniques, mathematical algebra, and constrained orthogonal function theory.

8. Acknowledgement

The first author would like to thank Pamela Das for her many thoughtful comments during the experiments and suggestions to make this manuscript clearer for all readers.

9. References

[9] TMS320C5402 Hardware Board, Texas Instruments, Dallas, Texas, 2001

Figure 9. Received fm symbol