CPM Neural Network Based Receiver for LEO Satellites

César Benavente[(1)], Ángel Martínez[(2)], Miguel A. Muñoz

[(1)]EUIT Telecomunicación. Universidad Politécnica de Madrid.
Ctra. Valencia, km. 7. 28031 Madrid, SPAIN
Email: cbpeces@euitt.upm.es
URL: http://welcome.to/cesarbenavente
Phone: +34 91 336 7830 Fax.: + 34 91 336 7832

[(2)] Instituto Nacional de Técnica Aeroespacial (INTA)
Ctra. Torrejón-Ajalvir, km. 4. 28850 Torrejón de Ardoz, SPAIN
Email: martinezja@inta.es
Phone: +34 91 520 1655

Abstract.

In this paper it is described the development of a receiver for GMSK signals based on the use of neural networks. Two cases are described: one in which the linear approximation of the GMSK signal is employed and other where phase changes are detected. Wide implementation, training and result differences are between both methods and they are described in the paper. The first approach is simpler than the second one, has an easier training but worst results from the point of view of error probability.

1. Introduction.

The use of neural networks has been widely extended in many science and engineering fields. One of the basic applications in which a neural network can be used is receiver data classification.

As an extension of the idea mentioned above we can think about the use of neural networks as a classifier of the samples of a received signal, in such a way that the classification allows identifying the transmitted data.

In this sense we can think of the constellation of a digital modulation. When we detect the transmitted data, the decision is based on the constellation properties. Depending on the modulation scheme, the constellation offers the transmitted data or the transition phase due to the transmitted data. Neural networks may be applied to both cases.

The development that will be described is based on the use of neural networks to classify the samples of the received signal. In concrete the receiver is thought for GMSK signals, and the classification allows extracting the transmitted data. GMSK modulation is a particular case of CPM modulations.

The development presented in the paper has been applied to a GMSK signal whose modulation parameters were BT=0.3 and modulation index 0.5, but may be extended to other parameter values or modulation schemes.

Our development assumes that the receiver is performing a perfect carrier and bit synchronisation.

CPM (continuous phase modulation) is a modulation scheme applied in satellite communications in which data determines the phase changes of the modulated signal. From this point of view, we have that the constellation points are determined by phase variations and the knowledge of the transition from one point of the constellation to another one gives the transmitted data.

A second point of view is obtained when a linear representation of the CPM signal is used to describe the modulated signal in order to simplify the receiver. In this case, it can be proved[1,2] that we can take decisions on the points of the constellation that will give the transmitted data.

Both cases may be solved applying neural network theory, but in different ways and with different structures. The use of neural networks for the decision task may offer some advantages over traditional methods.

In the case of the determination of the transmitted data based on phase transitions, data can be determined from the knowledge of the samples of the signal (for the in-phase and in-quadrature components) at the present decision instant and the precedent decision instant. This is due to the fact that only two possible phase transitions are allowed from a given point of the

constellation. In this case we need some memory to perform the decisions.

If the linear representation of the CPM signal is used, decisions are taken directly on the present constellation point, deciding values on both the in-phase and in-quadrature channels that will give the transmitted data. This operation may be performed by a simple hard decision.

The organisation of the paper is as follows. In the section 2 the basic concepts about the GMSK modulation are introduced, describing very briefly its exact description as well as the linear approximation formulation. Section 3 describes the receivers obtained when using neural networks in the implementation. Section 4 presents the results obtained from simulations approaching real conditions. Finally, section 5 describes the conclusions obtained from the development shown in the paper.

2. Basic Definitions.

The expression that defines a CPM signal is given by the equation:

$$x(nT_s) = e^{jf(nT_s)} \tag{1}$$

where:

$$f(nT_s) = ph\sum_{k=-\infty}^{N} a_k q(nT_s - kT) \tag{2}$$

is the instantaneous phase of the signal, whose value depends on the transmitted data. The data are referred as a_k and may take the values {+1,-1}, h is the modulation index and $q(nT_s)$ is the function that determines how the signal phase varies during a symbol period and whose properties distinguish the different types of continuous phase modulations (CPM). In the particular case of Gaussian Minimum Shift Keying (GMSK), that sharpener function is the integral of a Gaussian function and this is because the particular name.

Figure 1 shows the phase sharpener function $q(t)$ for a GMSK signal. Figure 2 shows the function $g(t)$ whose integral is the function $q(t)$.

Note that the function shown in figure 2 has finite duration and so $g(t)$ only varies into a finite duration interval. This is a practical approach because otherwise the system becomes too complex due to memory constraints. The function $q(t)$ saturates beyond a certain instant. The saturation value is 1, although if we consider strictly the truncated Gaussian function shown in figure 2 the saturation value is slightly lower than 1

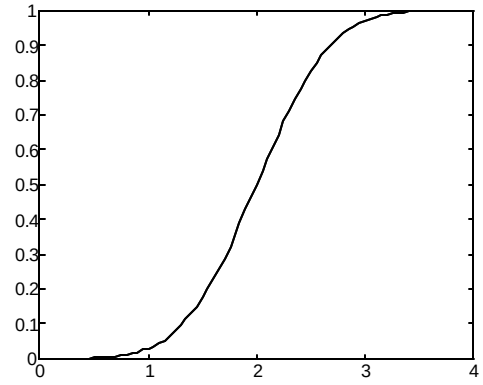

Figure 1. Function q(t).

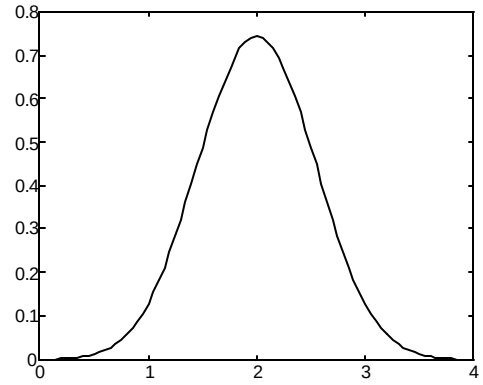

Figure 2. Function q(t).

The signal given by (1) may be separated into its in-phase component (the real part) and its in-quadrature component (the imaginary part), that will be used in the signal processing. The constellation of the signal and a time representation of the components are shown below.

The receiver used in the presented implementation is whole digital and two analog to digital converters are required (one per signal component) in order to obtain the samples of the signal that will be processed digitally on a DSP.

The oversampling factor employed at the ADC's depends on the functionality of the digital processing and the tasks to be performed. In our case, an oversampling factor of 2 is used. The signal is filtered and downsampled, obtaining finally one sample per symbol for detection tasks.

On the other hand we have that the linear approximation of the GMSK modulated signal[1] is given by:

$$x(nT_s) = \sum_k a_k p(nT_s - kT) \tag{3}$$

where a_k are the complex data and are directly related with the data α_k given in the exact representation by the equation (2)[1]. This data relationship is simply like a coding operation.

Data in equation (3) are complex and may take the values {+1,-1,+j,-j}, and $p(nT_s)$ is the basic pulse defined by the linear approximation, which determines an approximated expression of the GMSK signal as an amplitude modulated pulses signal (PAM).

Performing a slight modification of the modulator-demodulator structure[1] it is possible to obtain a system where the data in the received signal given by (3) are directly those we want to detect at the receiver. This modification involves including a coder before the modulator at the transmitter. This operation simplifies the receiver structure [1,2], whose detector may be simply a sign detector for both components.

Figure 3 shows the basic pulse $p(t)$ involved in the approximate description of the GMSK signal as an amplitude pulse modulated signal.

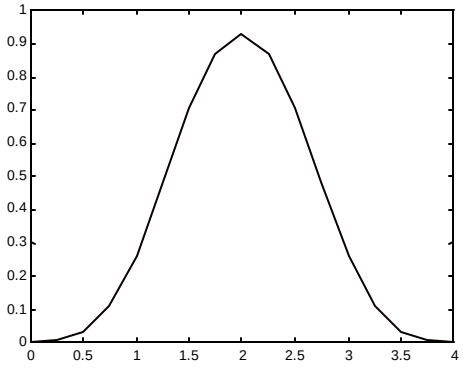

Figure 3. Function q(t).

Given the GMSK signal characteristics described above, the modulation constellation corresponding to the related signal is that represented in figure 4 where we can see several points due to the partial response characteristic of the modulation.

Due to the modulation features (continuous phase), phase changes from one data to the next one are not 90º, as they are produced in QPSK modulations (look at figure 5); but QPSK has less spectral efficiency than GMSK modulation..

Consider that from the point of view of the proposed detection the constellation presented in figure 5 corresponds to the 'ideal' case in which phase shift between one symbol and the next one is 90º.

In this case it is only needed at the receiver the detection of the sign of the signals at the in-phase and in-quadrature components in order the extract the transmitted information

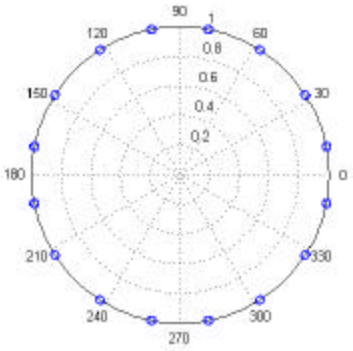

Figure 4. GMSK signal constellation.

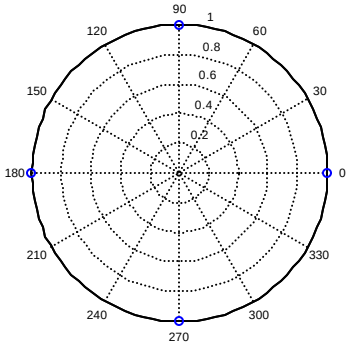

Figure 5. 'Ideal' constellation.

Figure 6 and 7 represent the in-phase and in-quadrature components respectively of a GMSK signal with the modulation parameters described above. This modulation produces a constant envelope signal as was shown in the constellation represented in figure 4. This property is quite important when high efficient amplifiers are employed at the transmitter because their non-linearity.

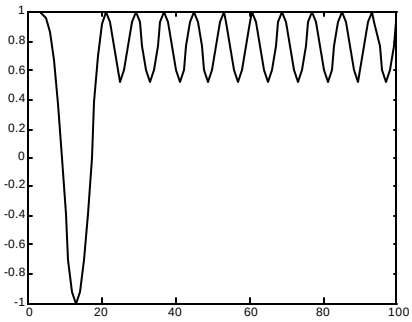

Figure 6. GMSK signal: in-phase component.

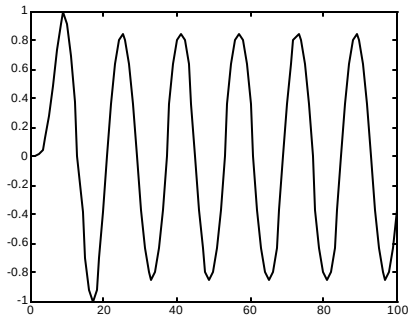

Figure 7. GMSK signal: in-quadrature component.

3. Neural Networks Receivers.

The structures corresponding to the receivers that use neural networks for data detection are shown in figures 8 and 11. The description of both networks is developed below. First we will describe the phase change detection based receiver.

3.1. Phase Change Detector Receiver: Neural Network Architecture.

Figure 8 shows the structure of the neural network in the case in which data detection is based on the classification of phase shifts. In this case two input elements per channel are required and they correspond to the sample at the present decision instant and the sample at the previous one. As the data to be decided are simply a_k defined in equation (2), we have two decision regions. This is due to the fact that given a decided data the next one is defined by a phase change of the order of 90º or –90º and a layer structure presented in figure 8 is enough for this purpose. The phase change is marked by the possible values of the data a_k for the exact representation.

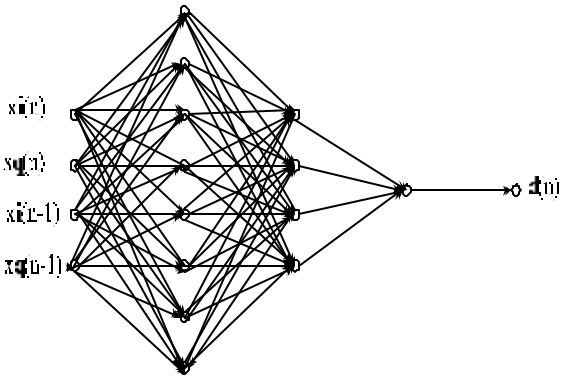

Figure 8. Neural network structure for the exact representation.

As can be seen in the figure, the neural network is composed of one input, 3 hidden layers and an output.

The input has 4 samples (elements), two of them are the samples of the in-phase and the in-quadrature components at the present decision instant. The other two samples correspond to the same samples at the previous decision instant. This information is required for the detection task.

The three hidden layers are constituted by 8, 4 and 1 neurones respectively, as can be shown in figure 8. The output is composed of an only element: the decision.

Table I shows the weights corresponding to the lines from each sample input to every neurone in the first hidden layer.

Table II shows the information corresponding to the biases of the neurones in the first hidden layer.

Table III shows the weights of the lines going from each neurone in layer 1 to every neurone in layer 2. Biases of neurones in layer 2 are shown in table IV.

Table V shown the weights of the lines going from the four neurones of the hidden layer 2 to the neurone of the hidden layer 3.

Finally, table VI shows the bias corresponding to the neurone in hidden layer 3.

Neurone	Input sample			
	$x_i(n)$	$x_q(n)$	$x_i(n-1)$	$x_q(n-1)$
0	0.2795	1.5201	0.5717	-1.9590
1	-1.3009	-0.5452	1.1087	-0.6084
2	1.5033	-0.6349	-1.2559	-1.7118
3	-0.8768	1.3665	1.2596	0.2692
4	0.6393	-0.7871	1.2196	0.6968
5	1.7839	1.5080	1.3197	-1.7170
6	1.4358	-1.3949	1.6931	-0.8679
7	-0.6075	-0.0330	-0.5360	0.5950

Table I. Input weights.

Neurone	bias
0	-1.7639
1	0.9988
2	-1.5389
3	-0.8593
4	1.2567
5	1.5910
6	-1.3082
7	-2.6130

Table II. Hidden layer 1 biases.

Layer 1	Layer 2			
	0	1	2	3
0	1.7939	1.2431	-0.6356	-1.1810
1	3.1149	-0.3045	-1.4860	0.2380
2	3.0656	-0.5932	-1.0331	-2.0942
3	2.0926	-0.1868	-0.8347	1.0647
4	2.8444	-0.2515	-1.4438	0.4724
5	-2.7845	-0.1753	0.9161	2.6060
6	-3.2329	0.7193	1.7182	0.4282
7	0.4177	1.1226	-0.0077	-0.1011

Table III. Weights form layer 1 to layer 2.

The transfer function of the neurones in the first hidden layer is the called 'tansig' function, which is represented in figure 9.

The transfer function corresponding to the third hidden layer is the called 'logsig', which is represented in figure 10.

Neurone	Bias
0	0.9466
1	-0.9251
2	0.3636
3	-1.0973

Table IV. Hidden layer 2 biases.

Neurones	0
0	15.4489
1	0.0197
2	5.2838
3	1.1401

Table V. Weights form layer 2 to layer 3.

Neurones	Bias
0	0.8692

Table VI. Hidden layer 1 bias.

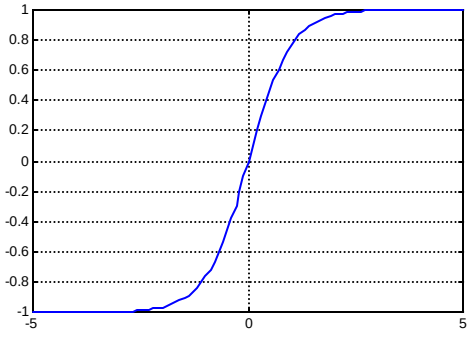

Figure 9. Tansig function.

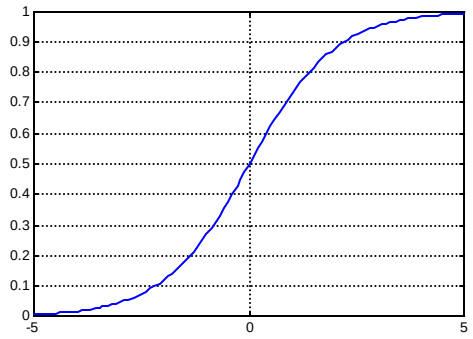

Figure 10. Logsig function.

3.2. Direct Decision Based Detector: Neural Network Architecture.

Figure 4 represents the case in which the choice was to use a receiver based on the linear approximation. In this case, we have to decide among four possibilities for the data a_k given by the equation (3), that is, four decision regions are required to perform the detection. The neural network shown in figure 11 performs the decision.

The neural network has an input that is composed of two elements that correspond to the samples of the in-phase and the in-quadrature components at the decision instant.

In this case the neural network has only one hidden layer that is composed of two neurones. Each input element is connected to every neurone of the hidden layer.

The output of the neural network is composed of two elements that correspond to the decisions on each signal channel. The decision depends on the neurones transfer functions.

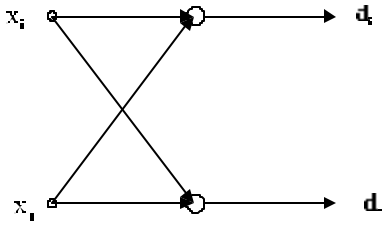

Figure 11 Neural network structure for the approximate expression.

Table VII shows the input weights of the network. By observation we can say that each neurone is specialised on one component of the signal.

Table VII shows the bias that affects each neurone of the layer.

Layer Neurone	Input element	
	$x_i(n)$	$x_q(n)$
0	33.3084	-25.6726
1	-29.5100	42.2367

Table VII. Input weights.

Layer neurone	Bias
0	0
1	2

Table VII. Input weights.

Figure 12 shows the transfer function of each neurone. This transfer function has been designed in order to meet the decision specifications and constellation characteristics. This is not a standard transfer function, it is a custom function.

Depending on the decision instant, we have to take the decision on the in-phase or the in-quadrature component of the signal. We are dealing with a detector where we should know if the present decision corresponds to one component or the other. This is a post-processing that allows to identify which of the two components of the output corresponds to the decision at the present decision instant.

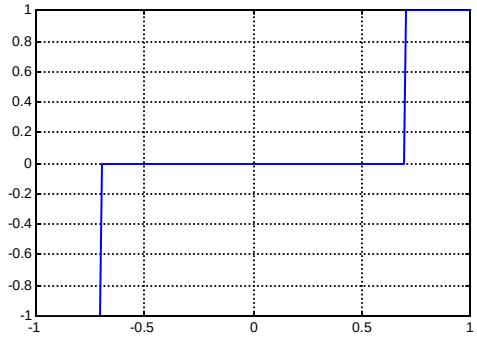

Figure 12. Designer transfer function.

4. Results.

Both neural networks have been trained with a modulated signal interfered by noise. Adaptation of the net weights has been performed taking into account the data to be detected as the target, the presence of noise and the modulated signal characteristics. If we introduce a distortion due to the transmission channel characteristics the nets will adapt taking into account that information about the channel.

Neural networks training is conditioned by the neurones transfer functions that constitute every layer.

Figure 13 shows the error probability obtained using the neural networks proposed in the paper.

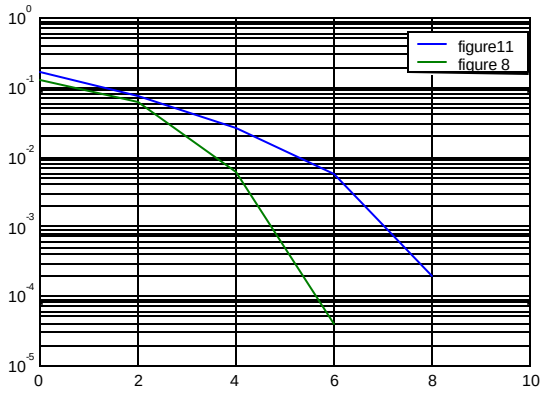

Figure 13. Detectors Error probability.

The training of the network states the behaviour of the network. Training with a signal non-corrupted by noise produces poorer error probability results than if we introduce a signal plus noise training pattern.

If we train the network with signals corrupted by noise produces satisfactory results, as it is shown in figure 13. It is not so important the signal choice as the conditions of the signal. We must train the network with similar signals to the ones present in real conditions.

It can be shown that the results for the first studied structure are better than that obtained from the second one.

5. Conclusions.

The structure of the neural network is simpler for the second case, where less information is required from the signal, but less information is given by the decisions.

The phase transitions detector is more reliable than the second structure. Some information can be added in the second case to improve the results. This information is the channel where the decision will be taken. Samples pre-processing may additionally improve the results.

LEO satellite channel is a very variant transmission path. Training sequence is used to obtain a proper performance of the system. This training sequence is used for different purposes (bit and carrier synchronisation acquisition) and also may be used to train the neural network. This allows to adapt the receiver along the satellite orbit.

From the results obtained from our work we conclude that the use of neural networks to solve problems as that stated in the paper produces satisfactory results without an important increase of the complexity of the receiver.

The use of neural networks may produce optimal solutions in the sense that we are able to adapt their properties to the signal received features at any time, that is, adapting to the channel conditions when there is a time variant channel.

Adaptive behaviour may be easily performed employing training sequences that permit to optimise the neural network weights, in such a way that continuously we can adapt the system and optimise the performance. This task increases the receiver complexity. Besides, the use of that training sequences are useful not only to that objective, they may be employed for the bit and carrier synchronisation tasks that can be performed digitally into a DSP[1] as well as every receiver task.

6. References.

1. Benavente, C. 'Linear receiver for CPM signal appropriate to digital implementation' ICSPAT'99, Orlando, Fl., November 1999.

2. Benavente, C. 'New implementations of bit and carrier synchronism in CPM modems', Doctoral Thesis, Universidad Politécnica de Madrid, 1999.