DIGITAL DESIGN OF A BROADCAST DVB-T REPEATER

DISEÑO DIGITAL DE UNA REPETIDORA DE EMISIÓN DVB-T

Evelio Astaiza Hoyos[1], Héctor Fabio Bermúdez Orozco[1].

[1] Engineering program of the University of Quindío, Grupo de Investigación en Telecomunicaciones de la Universidad del Quindío. eastaiza@uniquindio.edu.co, hfbermudez@uniquindio.edu.co.

Recibido: Marzo 5 de 2012
Aceptado: Mayo 28 de 2012
*Correspondencia del autor. Grupo de Investigación en Telecomunicaciones de la Universidad del Quindío, gituq@uniquindio.edu.co, tel 7460298 ext 108. Centro de Estudios e Investigaciones de la Facultad de Ingeniería – CEIFI Universidad del Quindío. Carrera 15 Nro. 12N, Armenia, Colombia

ABSTRACT

In this paper, an optimal signal processing design scheme is proposed for a digital broadcast DVB – T in Digital Terrestrial Television Broadcasting (DTTB) networks. The proposed scheme consists of two stages. In the first stage (input stage), a mixer performs a Frequency Down Conversion to translate the radio signal to IF band, then, an interpolator and decimator perform the processing to change the sample rate for the AFC block. The second stage (output stage) comprises one interpolator to change the sample rate at desired rate, and one mixer to return the signal frequency to RF band. Simulations show that the proposed scheme decreases the operations compared to the conventional design schemes. Additionally, implementation constraints are identified for each stage according to design requirements. Finally, some comments are presented in order to identify limitations and practical design issues in the proposed system.

Key words: DVB-T repeater, Frequency translation, Sample Rate Change, Digital Frequency Conversion.

INTRODUCTION

In this paper, we present a model to perform Digital Frequency Conversion and Sample Rate Change in a DVB-T broadcast network. Likewise, all proposed stages are analyzed and explained in detail.

Repeaters are signal emitters that are fed by a remote transmitter: they pick up the same digital or mobile TV signal that is directed at the end-customer receiver, amplify it and rebroadcast it towards more distant receivers.

The simplest repeaters are used for in-building coverage: an outside antenna picks up the transmitter signal, and it is then filtered and amplified for transmission inside the building. If the indoor transmitting antenna is sufficiently isolated from the outdoor receiving antenna, this simple scheme will work without creating feedback and oscillation of the entire chain.
Another type of repeater, which is not limited to indoor coverage, is a transposing repeater. In such a repeater, the input signal is shifted in frequency to a different channel for transmission with proper filtering, the frequency difference between output and input makes it possible to achieve the necessary isolation. (1)

In state-of-the-art repeaters, the received signal is usually sampled after downconversion to some intermediate frequency. Then it is digitally processed before undergoing RF upconversion, amplification, and retransmission.

Repeaters play a key role in broadcast network plan-

ning, providing coverage in those geographical areas where direct reception from the base station is not possible. They can retransmit either on the same frequency (gap fillers) or in a different channel (transposers), typically boosting signal levels by several tens of dB. Last-generation broadcast services contemplating the use of repeaters include both terrestrial and satellite OFDM-based networks such as DVB-T, DVB-H and DVB-SH. (2)

A block diagram of an input stage and output stage are shown on Fig 1 and Fig 2. The input stage comprises one mixer, one interpolator, and one decimator. The mixer performs a frequency downconversion to translate the radio signal to IF band. The interpolator and decimator perform the processing to change the sample rate for the AFC block. The output stage comprises one interpolator to change the sample rate at the desired rate, and one mixer to return the signal frequency to RF band. (3, 4)

Generally, input and output stages are implemented using direct interpolation and decimation. This guarantees an adequate signal processing but sacrificing computational efficiency. In this paper, we propose to implement these operations using polyphase filters in order to optimize the computation rate and identify practical design issues in proposed system. (5)

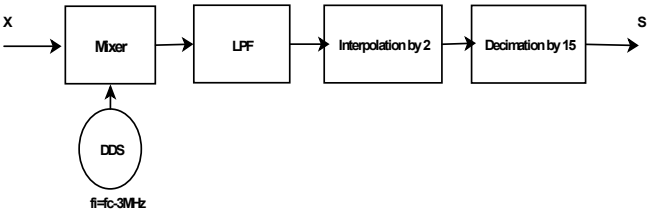

Figure 1: Basic structure of an input stage.

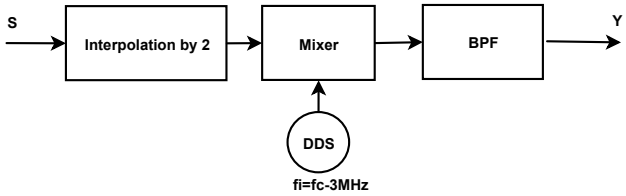

Figure 2: Basic structure of an output stage

The next sections describe how the Digital Frequency Conversion and Sample Rate Change in a DVB-T have been designed.

SYSTEM DESIGN

Input stage

The detailed block diagram is shown on Fig. 3. The mixer block uses a real multiplier and a DDS to generate the IF carrier at a frequency equal to FRF -3MHz. This block translates the spectrum at 3MHz. Then this signal is filtered with an equiripple low-pass filter for image rejection. This filter is designed with a cutoff frequency of 6MHz and 7MHz for the stop-band frequency with an attenuation of 40dB. In the worst case, the image can appear centered at 19MHz; thereafter, the signal is processed by a polyphase interpolator with an interpolation factor of 2 to convert the original sample rate of 90 Msamples/s to a sample rate of 180 Msamples/s. The interpolation filter is designed with cutoff frequency equal to ((fs/2)/I), stop-band frequency equal to 1.1((fs/2)/I), and an attenuation of 40dB in the stop band. In this stage, the signal is finally passed to a polyphase decimator to get down the sample rate by 15, converting the sample rate from 180 Msamples/s to 12 Msamples/s. The decimation filter is designed with a cutoff frequency equal to ((fs/2)/Q), a stop-band frequency equal to 1.1((fs/2)/Q), and an attenuation of 40dB at the stop band.

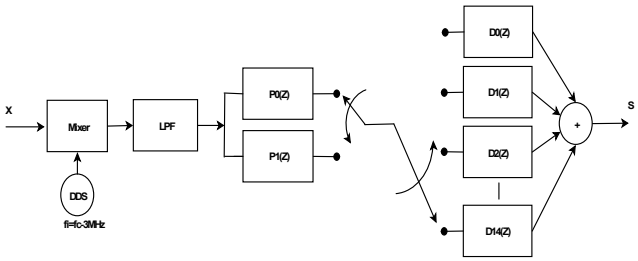

Figure 3: Detailed block diagram of the input stage

Output stage

The detailed block diagram is shown on Fig. 4. The first block is a polyphase interpolator to get up the sample rate by 8, converting the input sample rate from 12 Msamples/s to a sample rate of 96 Msamples/s. The interpolation filter is designed with cutoff frequency equal to ((fs/2)/I), stop-band frequency equal to 1.1((fs/2)/I), and an attenuation of 40dB at the stop band. And then, the mixer block uses a real multiplier and a DDS to generate the IF carrier at a frequency FRF -3MHz. Again, this block translates the spectrum at the 35MHz – 37MHz range.

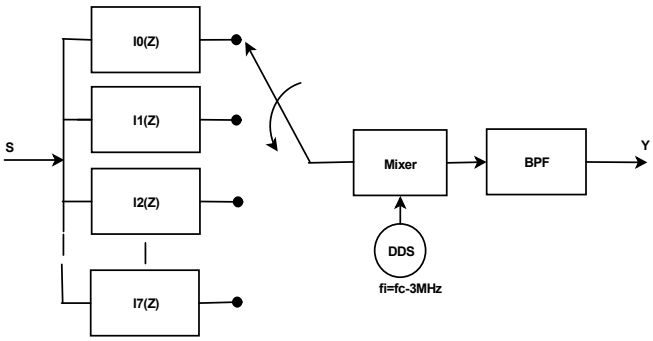

Figure 4: Detailed block diagram of the output stage

MATERIALS AND METHODS

To implement the system comprises the following elements:

Input stage

In this stage, the DDS can be implemented with a Cordic DDS as described in (Harris, 2007). To simplify the simulation of our implementation, we consider that the CORDIC DDS output is approximately equal to the carrier signal generated by the cos(x) function in Matlab, In this way, the number of operations performed by this block, including the operations performed by the gain and phase correction, are 11 complex multiplications and 5 complex additions. Afterwards, it is necessary to implement the image-rejection filter. This filter is designed with the fdatool, in which its parameters and transfer function are shown on Table 1.

Table 1: Parameters for image reject filter.

Response Type	Lowpass
Design Method	FIR Equripple
Filter Order	Minimum Order
Fs (MHz)	90
Fpass (MHz)	6
Fstop (MHz)	7
Apass (dB)	1
Astop (dB)	40

The resultant filter is a 128-taps image-rejection filter. This filter involves 128 operations. However, using polyphase decomposition, the number of operations are 64 per concurrent polyphase component. This number of operations can be also obtained if a symmetric FIR filter implementation is used.

The following Matlab code implements this block.

```
% Time base
time = length(x)*Ts; t = Ts:Ts:time;
fi=(fc-3e6);
%Mixing
y=cos(2*pi*fi*t);
x_3MHz1=x.*y;
x_3MHz=filter(Num,1,x_3MHz1);
```

For the image-rejection filter and all other filters implemented in these stages, we use equiripple filters because it is the method that enables compliance ripple and attenuation requirements.
The interpolator block is designed as a polyphase filter with 2 polyphase components. This design is created with the fdatool using the parameters shown on Table. 2.

Table 2: Parameters for the input stage interpolation filter.

Response Type	Lowpass
Design Method	FIR Equripple
Filter Order	Minimum Order
Fs (MHz)	90
Fpass (MHz)	22.5
Fstop (MHz)	25
Apass (dB)	1
Astop (dB)	40
Response Type	Interpolator
Interpolator Factor	2

The resultant filter is a 51-taps interpolation filter, in which the number of required operations is 51. Again, using the polyphase decomposition, the number of operations can be reduced to 26 per concurrent polyphase component.

The following Matlab code implements this block.

```
%Polyphase decomposition
p=polyphase(Hm);
p0=p(1,:);
p1=p(2,:);
%Polyphase interpolation
z0=filter(p0,1,x_3MHz);
z1=filter(p1,1,x_3MHz);
z=zeros(1,2*length(z0));
z(1:2:length(z))=z0(:);
z(2:2:length(z))=z1(:);
```

The decimator block is designed as a polyphase filter with 15 polyphase components. Its design is created with the fdatool using the parameters shown on Table 3.

Table 3: Parameters for the input stage decimation filter.

Response Type	Lowpass
Design Method	FIR Equripple
Filter Order	Minimum Order
Fs (MHz)	180
Fpass (MHz)	12
Fstop (MHz)	13
Apass (dB)	1
Astop (dB)	40
Response Type	Decimator
Interpolator Factor	15

The resultant filter is a 256-taps decimator filter, in which using the polyphase decomposition, the number of operations are 17 per concurrent polyphase component.

The following Matlab code implements this block.

```
%Input signal polyphase decomposition for decimation
d=polyphase(Hmdiez);
d0=d(1,:);
d1=d(2,:);
d2=d(3,:);
d3=d(4,:);
d4=d(5,:);
d5=d(6,:);
d6=d(7,:);
d7=d(8,:);
d8=d(9,:);
d9=d(10,:);
d10=d(11,:);
d11=d(12,:);
d12=d(13,:);
d13=d(14,:);
d14=d(15,:);
%Input signal polyphase decomposition for decimation
zd0=z(1:15:length(z));
zd1=z(2:15:length(z));
zd2=z(3:15:length(z));
zd3=z(4:15:length(z));
zd4=z(5:15:length(z));
zd5=z(6:15:length(z));
zd6=z(7:15:length(z));
zd7=z(8:15:length(z));
zd8=z(9:15:length(z));
zd9=z(10:15:length(z));
zd10=z(11:15:length(z));
zd11=z(12:15:length(z));
zd12=z(13:15:length(z));
zd13=z(14:15:length(z));
zd14=z(15:15:length(z));
%Decimation
w0=filter(d0,1,zd0);
w1=filter(d1,1,zd1);
w2=filter(d2,1,zd2);
w3=filter(d3,1,zd3);
w4=filter(d4,1,zd4);
w5=filter(d5,1,zd5);
w6=filter(d6,1,zd6);
w7=filter(d7,1,zd7);
w8=filter(d8,1,zd8);
w9=filter(d9,1,zd9);
w10=filter(d10,1,zd10);
w11=filter(d11,1,zd11);
w12=filter(d12,1,zd12);
w13=filter(d13,1,zd13);
w14=filter(d14,1,zd14);
w=w0+w1+w2+w3+w4+w5+w6+w7+w8+w9+w10+
w11+w12+w13+w14;
```

In which the signal w is the output of the input stage.

Output stage

The first block in this stage is the interpolator. It is designed as a polyphase filter with 8 polyphase components. Its design is created in the fdatool using the parameters shown on Table 4.

The resultant filter is a 972-taps image-rejection filter, in which using the polyphase decomposition, the number of operations are 122 per concurrent polyphase component.

Table 4: Parameters for the output stage interpolation filter.

Response Type	Lowpass
Design Method	FIR Equripple
Filter Order	Minimum Order
Fs (MHz)	96
Fpass (MHz)	6
Fstop (MHz)	6.14
Apass (dB)	1
Astop (dB)	40
Response Type	Interpolator
Interpolator Factor	8

The following Matlab code implements this block.

```
%Input signal polyphase decomposition for interpolation

I=polyphase(Hmintrx);
I0=I(1,:);
I1=I(2,:);
I2=I(3,:);
```

```
I3=I(4,:);
I4=I(5,:);
I5=I(6,:);
I6=I(7,:);
I7=I(8,:);
%Polyphase interpolation
SalInterp0=filter(I0,1,w);
SalInterp1=filter(I1,1,w);
SalInterp2=filter(I2,1,w);
SalInterp3=filter(I3,1,w);
SalInterp4=filter(I4,1,w);
SalInterp5=filter(I5,1,w);
SalInterp6=filter(I6,1,w);
SalInterp7=filter(I7,1,w);
SalidaInterp=zeros(1,8*length(SalInterp0));
SalidaInterp(1:8:length(SalidaInterp))=SalInterp0(:);
SalidaInterp(2:8:length(SalidaInterp))=SalInterp1(:);
SalidaInterp(3:8:length(SalidaInterp))=SalInterp2(:);
SalidaInterp(4:8:length(SalidaInterp))=SalInterp3(:);
SalidaInterp(5:8:length(SalidaInterp))=SalInterp4(:);
SalidaInterp(6:8:length(SalidaInterp))=SalInterp5(:);
SalidaInterp(7:8:length(SalidaInterp))=SalInterp6(:);
SalidaInterp(8:8:length(SalidaInterp))=SalInterp7(:);
```

As in the input stage, the DDS for the output stage can be implemented with a Cordic DDS. Again, to simplify the simulation in this implementation, we consider that the CORDIC DDS output is approximately equal to carrier signal generated by the cos(x) function in Matlab. In this way, the number of operations performed by this block, including the operations in the gain and phase correction, are 11 complex multiplications and 5 complex sums. Afterwards, the image-rejection filter is designed in the fdatool using the parameters and transfer function shown on Table 5.

Table 5: Parameters for the output stage image reject filter.

Response Type	Lowpass
Design Method	FIR Equripple
Filter Order	Minimum Order
Fs (MHz)	96
Fpass (MHz)	3
Fstop (MHz)	3.14
Apass (dB)	1
Astop (dB)	45

The resultant filter is a 1068-taps low pass band image-rejection filter. The number of operations to implement this filter is 1068, but using symmetric FIR implementation, the number of operations is 534. To maintain the linear phase characteristics of the filters, this filter does not involve polyphase decomposition.

The following Matlab code implements this block.

```
fs1=16*fs/15;
Ts1=1/fs1;
time1 = length(SalidaInterp)*Ts1; t = Ts1:Ts1:time1;
% base time
time2 = length(Num1); t2 = -fix(time2/2):1:fix(time2/2);
portadora=cos(2*pi*fi*t);
Num2=Num1.*cos(pi*(2*fc/fs1)*t2);
Salida1=SalidaInterp.*portadora;
SalidaMezclador=(1/(0.0045543))*filter(Num2,1,Salida1);
```

In which the correction factor (1/(0.0045543)) is used to compensate the gain offset introduced by the design that is calculated in the following MER script.

```
yn=simbI+i*simbQ;
thetaEstim=0.25*(angle((1/N)*sum(yn.^4))-pi)
gananciaEstim=sqrt((1/(N*mean(abs(constel).^2)))*sum(abs(yn).^2))
z=complex(simbI,simbQ);
figure(4); plot( z , 'o');
y=decide(z,constel);
figure(5); plot( y , 'o');
Rdata_I = real(y);
Rdata_Q = imag(y);
errI=Rdata_I-simbI;
errQ=Rdata_Q-simbQ;
data_Icuad=Rdata_I.^2;
data_Qcuad=Rdata_Q.^2;
errIcuad=errI.^2;
errQcuad=errQ.^2;
MER2=10*log10(sum(data_Icuad+data_Qcuad)/sum(errIcuad+errQcuad))
```

RESULTS AND DISCUSSION

To test the proposed processing, we generated three sets of data with IF values of 35MHz, 36MHz and 37MHz at 90 Msamples/s using 16-QAM modulation with a raised cosine pulse shaping filter. These signals are processed by the input stage, and the output stage in a script that implements the receptor. The receptor was synchronized considering the filters delays, implementing the phase corrections and selecting the best sample that maximize the MER.

On Fig 5, it is shown the power spectral density of the test 16 QAM passband signal at repeater input, in which we can observe it is centered at 35MHZ frequency carrier with a 6MHz bandwidth.

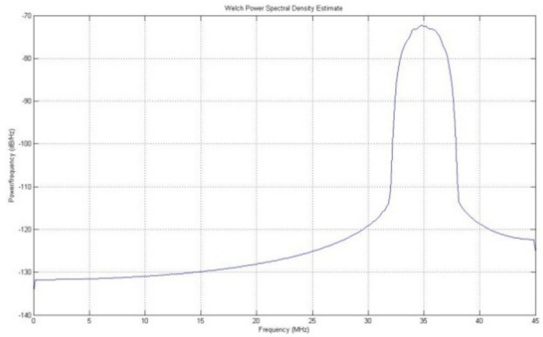

Figure 5: Power spectral density of the input signal. .

On Fig 6, it is shown the power spectral density of down-converted and filtered test input signal where we can observe that signal was translated to 3MHz central frequency, and the image frequency placed at 23MHz is around 40dB below.

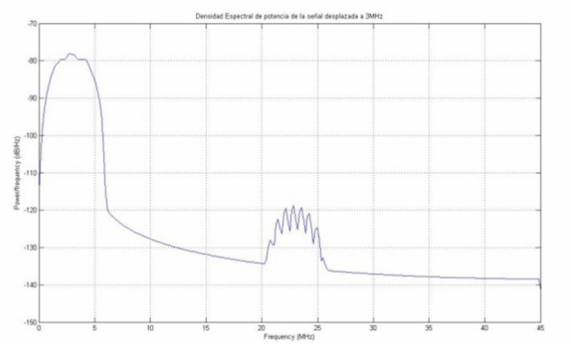

Figure 6: Power spectral density of down-converted and filtered signal.

On Fig. 7, it is shown the power spectral density of the signal interpolated by the input stage, in which we can observe that signal rate is increased by interpolation factor 2, and some components produced by upsampling process around 87MHz and fs=180MHz appear.

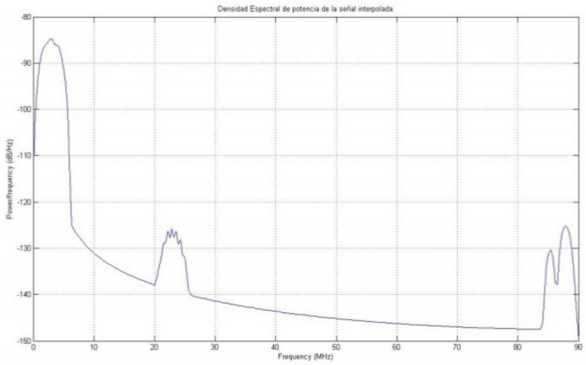

Figure 7: Power spectral density of the signal interpolated by the input stage

On Fig. 8, it is shown the power spectral density of the signal decimated signal in the input stage, in which we can observe that signal rate is decreased by decimation factor 15 where the signal bandwidth is 6MHz and fs=12MHz.

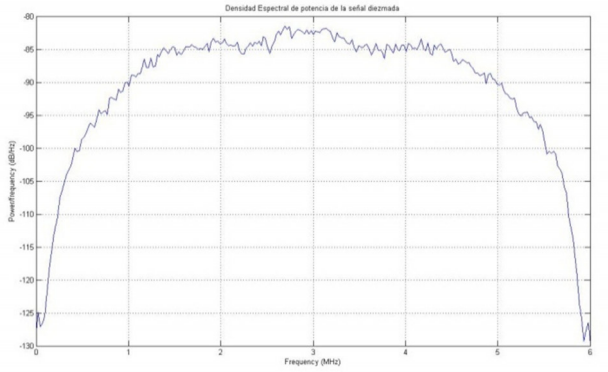

Figure 8: Power spectral density of the decimated signal in the input stage.

On Fig. 9, it is shown the power spectral density of the of the interpolated signal at the output stage, in which we can observe that signal rate is newly decreased by interpolation factor 8 where the signal bandwidth is 6MHz and fs=96MHz, and some components appear produced by upsampling process around 9MHz, 15MHz, 21MHz, 27MHz, 33MHz, 39MHz and 45 MHz.

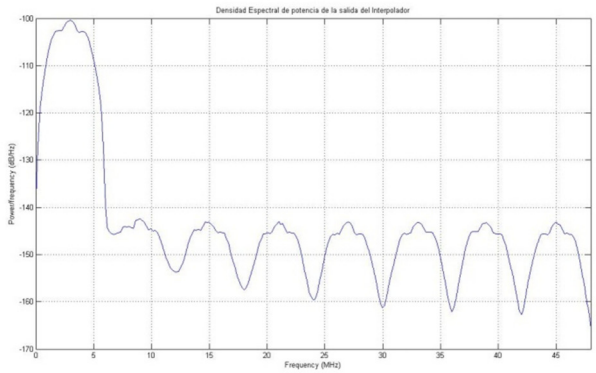

Figure 9: Power spectral density of the interpolated signal at the output stage.

On Fig. 10, it is shown the power spectral density of the up-converted and filtered signal at the output stage, in which we can observe that signal rate is 96MHz, and the carrier frequency is 35MHz again; in the same way, It is observed that the SNR is around 45dB.

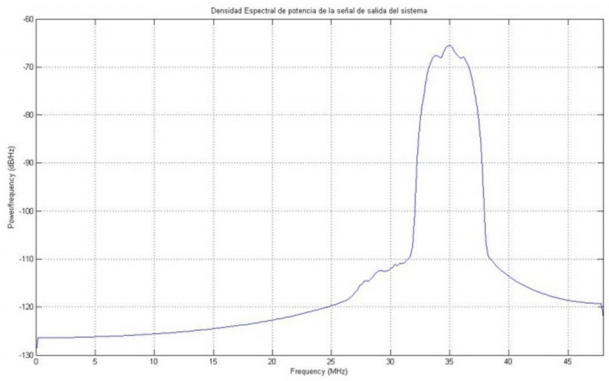

Figure 10: Power spectral density of the up-converted and filtered signal.

Finally, on Fig. 11, it is shown the Received constellation in which we can observe that symbols are affected for noise produced by dispersion produced in filter stages, and then compensated in an equalization and synchronization stages in the receptor.

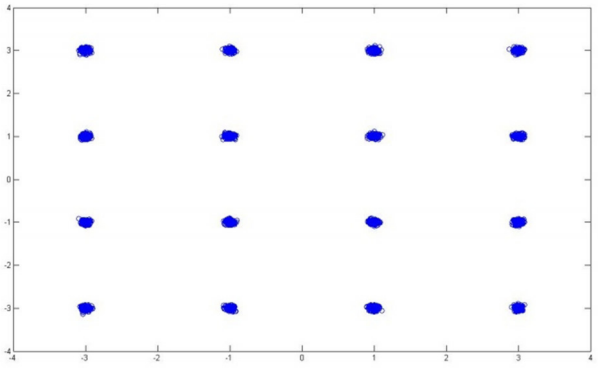

Figure 11: Received constellation

The obtained MER values are presented in the Table 6.

Table 6: MER per carrier frequency

IF (MHz)	MER(dB)
35MHz	36.5301
36MHz	36.4023
37MHz	36.5182

CONSIDERATIONS

In the input stage, we present a practical solution for Digital Frequency Conversion and Filtering Stages for a Broadcast Repeater. In the repeater, the image-rejection filter can be suppressed to save some operations. However, the removal of this filter introduces spectral distortion in the interest band. Likewise, in the output stage, the image-rejection filter had been designed like a low pass filter with an 3,14MHz bandwidth, and then its frequency response is translated to fRF to meet independence on IF frequency requirement.

Summarizing, the amount of operations required by this design is shown in the Table 7.

Table 6: Operations required by this design

Stage	Block	Number Operations
Input stage	Down converter	76
Input stage	Interpolator	26
Input stage	Decimator	17
Output stage	Interpolator	122
Output stage	Up converter	546
Total Operations		**787**

In the design proposed, an important issue is the dispersion produced by the filtering stages, this dispersion on constellation implies a MER decrease that must be compensated in reception.

CONCLUSIONS

In this report, we have presented improvements by efficient signal processing and SDR technologies applied to communications systems. In the same way, we have presented a general design to perform Digital Frequency Conversion and Filtering Stages for a Broadcast Repeater using this technology. By means of block diagrams, every software component has been described. Finally, the prime implementation considerations have been discussed.

As a final comment, we consider SDR and its evolution, Cognitive Radio, as a promising technology, which could contribute to improve communications systems. However, its implementation on real systems is going to take some time since the current programmable logic devices do not have enough processing capacity to implement the algorithms.

BIBLIOGRAPHY

1. Harris, F. Ultra Low Phase Noise DSP Oscillator. IEEE Signal Processing Magazine, DSP Tips and Tricks. 2007.
2. Morlet, C., Boucheret, M.-L., Calmettes, V., Paillassa, B., & Perennou, T. Towards Generic Satellite Payloads: Software Radio. Parallel and Distributed Processing International Symposium, 2003, pág. 7.
3. Pavel, K., & Vejrazka, F. Software Radio and its Applications in GNSS. 46th International Symposium Electronics in Marine, ELMAR-2004. Zadar Croatia, 2004.
4. Buracchini, E. The Software Radio Concept. IEEE Communications Magazine, 2000.
5. Tang, H. Some Physical Layer Issues of Wide-Band Cognitive Radio Systems. First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005; 151-159.