

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/247957486>

# NOAA Signal Decoding And Image Processing Using GNU-Radio

Article · January 2013

CITATIONS

3

READS

9,059

2 authors:



**Nicolae Crişan**

Universitatea Tehnica Cluj-Napoca

65 PUBLICATIONS 90 CITATIONS

[SEE PROFILE](#)



**Ligia Cremene**

Universitatea Tehnica Cluj-Napoca

56 PUBLICATIONS 144 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Spectrum Sharing - Game Theoretical Approach [View project](#)



Future WiFi [View project](#)

## NOAA SIGNAL DECODING AND IMAGE PROCESSING USING GNU-RADIO

*Nicolae Crisan, Ligia Cremene*  
Technical University of Cluj-Napoca  
E-mail: nicolae.crisan@com.utcluj.ro  
Phone: 0264401241

**Abstract:** The paper presents several technical issues and software implementations proposed in order to decode, process, and render images received from NOAA satellites. NOAA satellites are seeking extreme weather events on Earth. The proposed solutions offer the possibility of decoding and processing the NOAA signals, by using the PC and its audio-in port, without any extra signal processor-based boards. Also, system architecture is open, according to the GNU requirements and hardware independent, due to its capability of working offline, by decoding a pre-recorded .wav file. Finally, the results are converted to a standard image file that lends itself easily to processing for relevant feature extraction. In spite of its hardware independency, the algorithm is capable to operate also online with SDR receivers. These receivers must follow the Universal Software Radio Peripheral specifications on USB ports.

**Keywords:** *Software Defined Radio (SDR), Universal Software Radio Peripheral (USRP), GNU radio.*

### I. INTRODUCTION

GNU radio [1] is an open development toolkit that provides free processing components in order to implement software radio interfaces. GNU radio toolkits are primarily written in Python language based on Linux scripts and tools. Some performance critical signal processing routines are written in C++ using floating point processor extensions. Thus, a programmer can imagine simple and rapid applications still compatible with SDR receivers that follow the Universal Software Radio Peripheral (USRP) specifications.

USRP from Ettus research [2] enables users worldwide to address a wide range of research topics, opening the way through the software radio approach with the help of the GNU radio toolkit [3] which it is compatible with.

The Automatic Picture Transmission (APT) system [4] uses data streams to transmit a visible image during daylight using channel A and an infra-red image on channel B (all day and night long). The APT analog signal is transmitted continuously via NOAA-8 through NOAA-14 satellites. This signal was designed especially for inexpensive receivers and ground stations.

National Oceanic and Atmospheric Administration (NOAA) [5] is a US department that administrates an environmental satellite system that sends data from space in order to monitor and track weather change and extreme weather events. NOAA operates two types of satellites: geostationary satellites, for US monitoring, and polar-orbiting satellites that circle the planet and transmit pictures continuously from about 540 miles above the Earth. Transmitted data may also be used to calculate ocean temperature and for a better understanding of the Earth

climate change.

This paper focuses on the algorithm and software implementation in order to decode and process the APT signal using original routines. The hardware platform used for receiving and recording the APT signal in a .wav file is outside the topic of this paper. The main objective of this work is to reduce complexity by intensively using offline signal processing techniques and image processing in order to counteract the Doppler effects.

The paper is organized as follows. Section II overviews the main challenges in the reception of NOAA satellite images and presents the core ideas of the proposed solutions. Section III discusses the block-diagram for NOAA satellite reception and the D-APT-AM signal recording. Section IV describes the use of GNU radio companion to decode the APT-wav files. Section V presents the adjustments (processing) performed on the received image. Section VI presents the conclusions.

### II. PROBLEM FORMULATION

The present paper aims not to highlight the hardware platforms or routines involved in converting the APT analog signal into a .wav file. This is a common issue within the ham radio community, together with the open distribution of the APT-wav files [6].

For instance, a low-cost ground station uses a SDR receiver (SDR1000-6000) with a satellite track device entirely controlled by the computer [7]. This tracking device follows the NOAA-X satellites according to the telemetry data, assisted by software (see Fig. 1) [8]. Recorded .wav files are also available from other communities who share it freely via servers. The APT-wav files are available for download in 8 or 16 bits/sample, the

last ones having higher definition and offering a better quality. Fig. 1 captures an example of how Keplerian data together with software and internet are tracking the satellite 24 hours a day. The trajectory of the NOAA satellites is elliptic, so the distance varies with time.

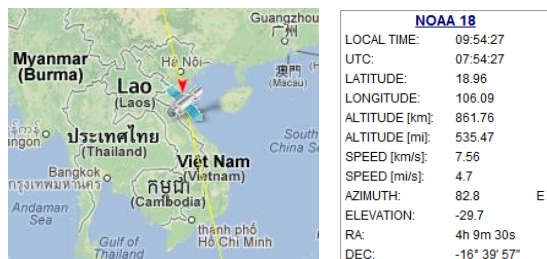


Figure 1. Online Keplerian data via Internet using tracking satellite software NOAA-18

The question that emerges is whether the .wav file can offer at least as much information as the decoded image, from Fig 2. The answer is no. The image in Fig. 2 is acquired with a professional hardware platform. This platform performs Doppler shift, frequency correction and line synchronization based on the Keplerian data extracted directly from the APT signal [4]. Most of the wav-APT signals are stored as they are making it possible to do these corrections offline, by software, providing the data without any change at the receiver level in terms of frequency auto-tuning or Doppler shift compensation.



Figure 2. NOAA-18 image with each line synchronization and Doppler shift compensation [6]

The wav-APT file becomes very interesting from this point of view for any programmer who is interested in intensively using software radio. The main goal of image and signal processing is in reducing the complexity of the necessary hardware platform. This means that the use of a very low-cost ground station becomes affordable for anyone who uses a SDR receiver and an omni-antenna. The tracking device is no longer necessary, at the cost of SNR loss. Of course, a noisier image requires more processing and the need for processing increasing.

Another challenge emerges from the wide use of the cheap narrow band transceivers with a narrower IF bandwidth (only 25 KHz instead of 34 KHz) as it is

required for the APT signal. This leads to the loss in image contrast. Fortunately, the reduced range of luminance along the black and white grayscale lends itself very well to histogram equalization techniques. The main concerns in the next sections are whether image processing techniques can cope with these problems.

### III. APT DATA FRAME FORMAT

The APT analog signal [4] is FM modulated using an RF carrier between 137.1 and 137.9125 MHz, depending on the index of the NOAA satellite. For example, NOAA-18 transmitter is tuned on 137.9125 MHz.

A low-cost ground station demodulates the FM signal and applies the FM demodulated signal to the PC-audio-in board. The APT-AM demodulated analog signal (Fig. 3) is a 2.4 KHz subcarrier in the range of the PC sound card. The D-APT-AM signal is sampled by the sound board and recorded as is on the hard-disk without any receiver adjustments in a .wav format.

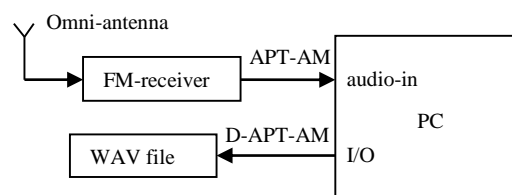


Figure 3. The most simple block-diagram for NOAA satellite reception and the D-APT-AM signal recording

The figure 4 shows the basic APT format. The transmitted carrier is FM modulated on 137.5 MHz and approaches 137.9125 MHz depending on the satellite index. An AM subcarrier on 2400 Hz modulates image data, as amplitude variation along a gray scale. Each word is sampled using 8 bits/pixel along one row, which lasts exactly 0.5 seconds (2 lines/second). The equivalent data rate is 4160 word/second with a Digital/Analog accuracy of 8 MSB's of each 10 bit word. For each image, only 909 words/row are useful data; the remaining ones are for synchronization and telemetry.

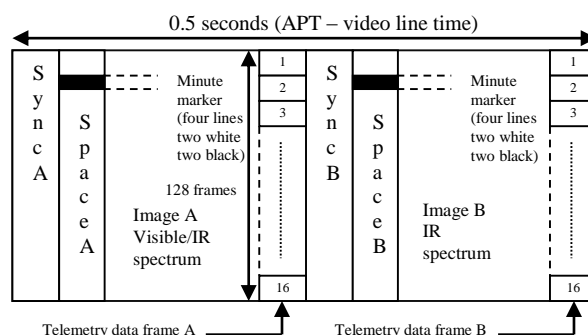


Figure 4. APT data frame format [4]

Sync A is a 1040 Hz square wave of seven cycles. Sync B is an 832 pulse train having also seven cycles [4]. The total number of pixels along a row is 2080, taking into account A and B images, Sync A and B and telemetry frames. There are 128 telemetry frames with 8 lines each. All those 128 frames put together a total of 1024 lines. The image resolution is 909/1024 pixels.

Additional explanations on Telemetry and Sync frames are not necessary at this point because we are not using any online adjustments based on these frames. These adjustments are performed offline, after the signal acquisition. The signal acquisition is running continuously, and the PC is saving data in a .wav file, according to the block diagram from Fig. 3.

Offline Doppler shift compensation and others adjustments will be the topic of Section V and are entirely software defined.

#### IV. USING GNU RADIO COMPANION TO DECODE THE APT-WAV FILES

The sampling rate of the recorded .wav file is 11.025 KHz. This sampling frequency depends especially on the technical performance of the sound board and may be even higher. Nevertheless the most notable thing is that the D-APT-AM is an AM 2.4 KHz signal.

After the reading of the .wav file (see Fig. 5) with the Wav File Source block, the signal is filtered with a low pass filter. The higher cut-off frequency of the filter upper bounds, the higher are the components at 2.4 KHz. The filtered signal is re-sampled to 9.6 KHz, which is four times higher than 2.4 KHz.

The *Rational Resampler* must interpolate the 9.6 KHz signal because it is not an integer multiplier of the input frequency (11.025 KHz). After the interpolation, the 9.6 KHz signal presents four samples for each word of the AM 2.4 KHz signal. So we have four samples/period and the samples come with 90 degrees phase shift (figure 5).

Using two consecutive samples we can detect the instantaneous amplitude of the AM 2.4 KHz signal at a rate of 4.8 KHz. The amplitude will be  $A = \sqrt{x^2 + y^2}$ , if every pair of two consecutive samples is converted to a complex number  $V = x + jy$ , where  $x$  and  $y$  are *float* variables and the block *Float to Complex* does this conversion. The use of the complex number  $V$  offers some advantages making it easier to calculate the value of amplitude  $A$ .

The block *Complex to Magnitude* acts like an AM demodulator and thus the signal caring the useful data is present at the output of the block. However, the demodulated signal is not yet ready for saving because the true symbol rate in APT format is 4.16 KHz. The next block interpolates the signal again to 4.16 KHz. The interpolated signal is now ready for saving, after a range expansion between 0 and 255. This range happens to be equal to the number we can render on using eight bits. Taking into account that the CAN's resolution is 8 bits/word this processing follows a logic choice. As a rule of thumb, it is mandatory to use a minimum number of bits for a word in a file. Pursuing this idea the next block converts *float* numbers to *unsigned char*. Next, one block detects the first Sync A synchronization burst of 1040 Hz in order to skip the first row of the image which is incomplete. Finally, the signal is saved as is, in a .dat file, by the last block.

All Radio Companion blocks are written in Python under the GNU Radio packages [3]. The block-diagram

presented in Fig. 5 is responsible only for reading, decoding and saving data in a file. The main challenge here is the .dat file which is not a compatible image format.

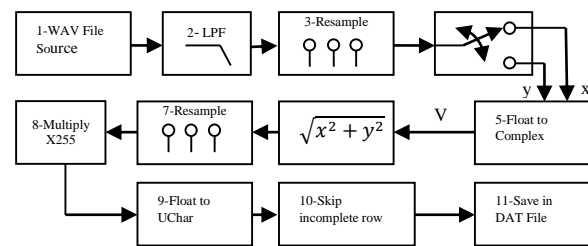


Figure 5. Block diagram of the second down-converter

This problem is avoided with the help of the *Image Magic* program installed also with the Linux packages. The conversion from .dat to .png image file is done by the following command from the Linux terminal:

```
convert -size 2080x1024 -depth 8 gray: output.dat output1.png
```

The converter puts together every 2080 words into a row, generating 1024 rows, according to the number of APT frames (see Fig. 4). The gray depth is 8 (0-255 levels) and the results are saved in a .png image format.

#### V. THE IMAGE ADJUSTMENTS

When the satellite approaches the ground station [9], travelling at about 7.5-7.6 Km/s, at an altitude of 700 Km above the Earth, the Doppler effect shifts the frequency with  $\pm 3$  KHz. At this point, the carrier frequency is higher than 137.9125 MHz with maximum 3 KHz, just when the relative velocity reaches a maximum pick.

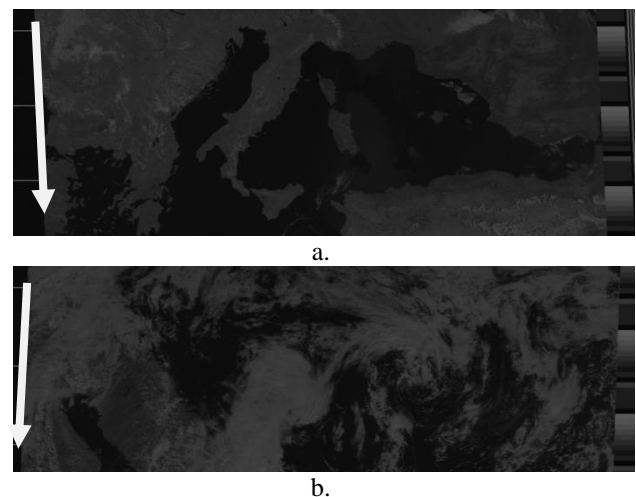


Figure 6. Row shift effect over the decoded image

The frequency drops suddenly with 6 KHz after the satellite passes over the view point, from front to back. This shift affects the data rate and also the geometry of the image. The effect is captured in Fig. 6. In Fig. 6.a the image is shifted to the right – when the satellite comes from the front. The same image is shifted to the left when the satellite is moving away from the reference view point (Fig. 6.b).

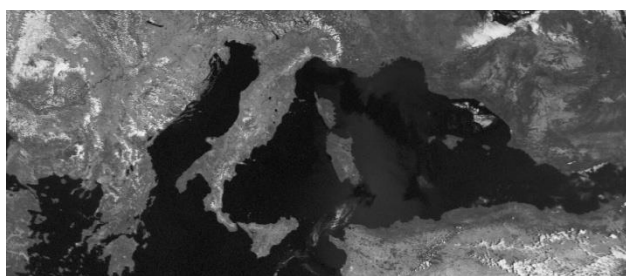


The correction process (image geometry compensation) – shown in Fig. 7.a – shifts each row in the opposite direction using a vertical line as reference in order to counteract the Doppler effect. The black space A region (see Fig. 4) is used to define this line and the transitions from black to gray at the beginning of each row of the image. In this way, the Doppler shift effect is compensated for each row entirely by software.

Another effect that needs correction at the level of the satellite could avoid the geometry distortions by maintaining almost the same resolution of 4 Km along the scan line [10][11][12]. This correction takes into account the instantaneous distance between the satellite and the earth [12] and its rotation speed. Fortunately, NOAA satellites make this correction online. Nevertheless all these undesired factors like earth rotation, earth curvature and Doppler shift effects are cumulative and all distort geometry of the image. This is shifted to the left or to the right and could be corrected by software following the rule of shifting the image lines in the opposite direction. From these factors, only the Doppler shift alters both the geometry and the luminance level of the image [6].



a.



b.

Figure 7. Image adjustments a. Geometric corrections  
b. Histogram equalization

The 25 KHz-channel transceiver limits the range of the image luminance along the gray scale. The use of a common transceiver with a 25 KHz FM band instead of the 34 KHz band decreases the image contrast. This loss increases with the increasing of the satellite velocity and with the carrier frequency. A 34 KHz receiver using a wider band could counteract better this effect. The best way to counteract the Doppler shift is to tune the transceiver using the telemetry data transmitted together with the image. Here the proposed method is compensating the

luminance loss depicted in Fig. 7.b by using a histogram equalization technique [8].

Histogram equalization is a common technique, broadly used in image processing every time an image suffers from a loss of contrast. For a dedicated transceiver with a 34 KHz FM band, 26% of its the power spectrum is outside the range of a commonly used transceiver. The loss in terms of the contrast is not 26% because the signal energy is not uniformly distributed along the spectrum. The loss affects only higher frequencies that have not as much occurrence probability like the lower frequencies. This leads to the conclusion that the depreciation of the contrast with 26% represents the worst case. However, this depends on the image content and affects more the brighter images. After the histogram equalization, the image becomes clearly visible with a significant gain in quality (Fig. 7.b).

## VI. CONCLUSIONS

A new method has been proposed for NOAA satellite image reception based on GNU radio technique. The procedure is based on a sequence of image processing procedures for restoring the distorted image, instead of using the more complicated frequency correction and line synchronization based on the Keplerian data extracted from the APT signal. The advantages of the proposed method are the accuracy and affordability, no professional hardware being needed.

## REFERENCES

- [1] Alexander M. Wyglinski, Maziar Nekovee, Y. Thomas Hou, *Cognitive Radio Communications - Principles and Practice*, Elsevier – Academic Press - British Library Catalog, pp 621-631, 2010, ISBN 978-0-12-374715-0
- [2] Documentation available about USRP – Universal Software Radio Peripheral is available at <http://www.ettus.com/>
- [3] Documentation about GNU radio is available at <http://gnuradio.org/redmine/projects/gnuradio/wiki/BuildGuide>
- [4] Documentation about APT – Automatic Picture Transmission is available at <http://www2.ncdc.noaa.gov/docs/klm/html/c4/sec4-2.htm>
- [5] Information about NOAA – The National Oceanic and Atmospheric Administration is available at <http://www.noaa.gov/satellites.html>
- [6] The .wav file used in the paper is available at [http://www.fredvandenbosch.nl/satellites\\_WAV.html](http://www.fredvandenbosch.nl/satellites_WAV.html)
- [7] A list of low cost SDRs capable of receiving and decoding APT signals is available at <http://www.flex-radio.com/>
- [8] A software tool for real time satellites tracking is available at <http://www.n2yo.com/satellites/?c=4>
- [9] T. Langdon, *A close up on Doppler Shift*, The AMSAT Journal, 2010
- [10] Bate, Roger; Mueller, Donald; White, Jerry, *Fundamentals of Astrodynamics*, Dover Publications, Inc., New York, 1971, ISBN 0-486-60061-0.
- [11] Acharya and Ray, *Image Processing: Principles and Applications*, Wiley-Interscience 2005 ISBN 0-471-71998-6
- [12] John, A. Richard, *Remote Sensing Digital Image Analysis*, Springer – New-York Dordrecht London, pp 56-62, 2013, ISBN 978-3-642-30062-2

