EE351 Communication Systems Lab

Live Video Stream using GNU Radio and USRP B200

Pintu Kumar (11640650) *Rudrashish Pal (11640870)*
Sharad Roy (11640920) *Siddharth Jain (11640990)*

Objective

In this project we aim to establish a video transmission system that streams in real time from a video camera i.e. transmit a real-time video from one station to the other. We have used GNU Radio, USRP B200 and VLC Media Player for this purpose. In this report we describe the techniques used in this experiment and the related theory.

Gaussian-filtered Minimum Shift Keying

Gaussian-filtered Minimum Shift Keying (GMSK) modulation is type of continuous-phase frequency-shift keying i.e. there are no phase discontinuities. GMSK differs from MSK (Minimum Shift Keying) in that a Gaussian Filter of an appropriate bandwidth (defined by the *BT* product) is used to shape the data stream before the modulation stage. This reduces sideband power, which in turn reduces inter-band interference between high frequency carriers in adjacent frequency channels. The time-domain impulse response of the filter is described as follows -

$$h(t) = \frac{k_1 B}{\sqrt{\pi}} e^{-k_1^2 B^2 t^2} \qquad \ldots (1)$$

Here, $k_1 = \frac{\pi}{\sqrt{2ln2}}$ and B is half-power bandwidth. GMSK modulation has improved spectral efficiency when compared to other phase shift keyed modes. A further advantage is the fact that it is more resilient to noise, as most noise is mainly due to amplitude variations (no amplitude variations are used in GMSK modulation).

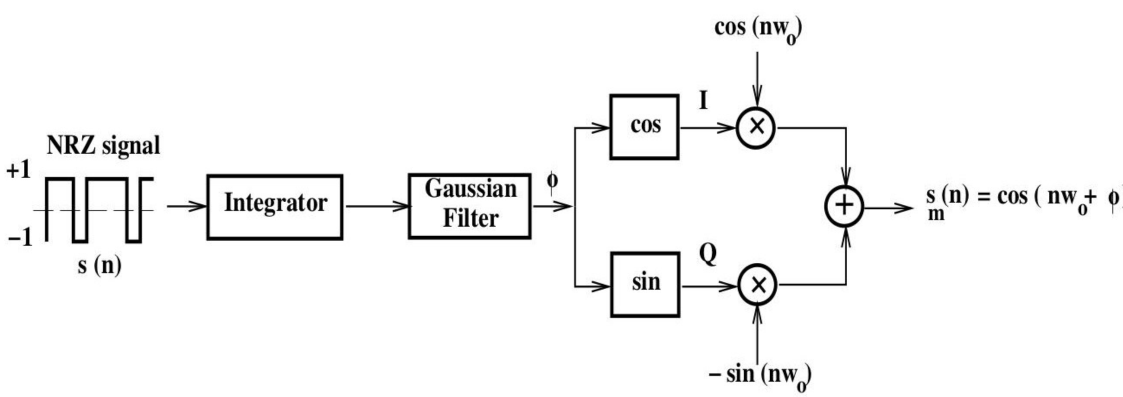


Figure 1 *Representative block diagram for GMSK modulation*

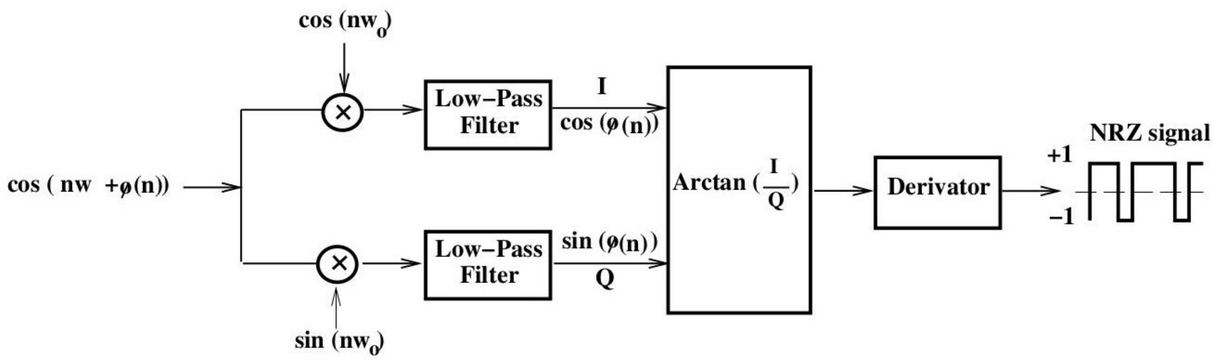
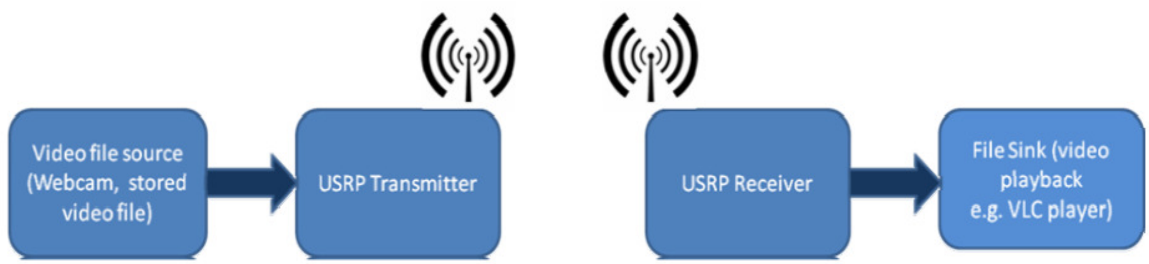


Figure 2 *Representative block diagram for GMSK demodulation*

Basic Block Diagram



User Datagram Protocol

User Datagram Protocol (UDP) is a networking protocol used by programs/applications running on different computers on a network to transmit/receive data packets. A strong advantage of UDP is that it does not require prior communication to set up data channels between the transmitting and receiving stations and is more efficient in terms of bandwidth. UDP uses a simple connectionless communication model with a minimum of protocol mechanism. The first eight bytes of a datagram contain header information, while the remaining bytes contain message data. A UDP datagram header contains four fields of two bytes each: source port number, destination port number, datagram size, checksum. In this experiment we use UDP to setup a communication interface between VLC Media Player and GNU Radio.

Video Streaming

VLC Media Player is an application which outputs real-time video. We have used an inbuilt webcam at transmitter station as a video recorder. The application takes the real-time video stream and converts it to a format suitable for continuous data transmission and reception by GNU Radio. This video format is known as **transcoded stream (.ts)**. The process of video transcoding is normally a two-step process. The first part of the process is decoding. This is whereby the original data (live video in this case) is transferred to an uncompressed format. The second part of the process is the re-encoding whereby the data will now be transferred to the host application (GNU Radio) in the desired format.

The transcoded stream is transmitted using a UDP communication interface. This basically sets up a ‘pipe’ for continuous transmission of data packets. The UDP pipe is essentially the link for data transfer from VLC Media Player to GNU Radio. We implemented the communication interface in the following steps.

1. Open **VLC Media Player** and click on **File** followed by **Stream**.
2. In the dialogue box that shows up, select **Capture Device**. This specifies the application that the source of video stream is a external recording device.
3. Select **Video camera** as **Capture mode**. **Video device name** will set to **/dev/video0** by default which specifies the inbuilt webcam as the video source. Similarly, **Audio device name** will be set to **hw:1,0**. If a device other than the default is to used, set the above parameters accordingly, specifying the device name.
4. Set **Video standard** to **All** from the drop-down menu. Clicking on **Next** will show the **Source** as **v4l2:///dev/video0**. Continue by clicking on **Next**.
5. In the **New destination** drop down menu, select **UDP (Legacy)** and click on **Add**. In the tab that shows specify the address to which the video stream data is to be sent (such as 127.0.0.1). The **Port** is set to **1234** by default. Select **Next**.
6. In the tab that opens, select the checkbox to **Activate Transcoding**. From the drop-down menu adjacent to **Profile** select **Video - MPEG-2 + MPGA (TS)**. Click **Next**.
7. In the following tab, click on **Stream**. The webcam/external recording device should be activated and VLC should be streaming live video.

GNU Radio Transmission and Reception

In GNU Radio we used the following flow diagrams on receiver and transmitter stations to implement GMSK modulation on the incoming data packets, transmit using USRP, receive and demodulate the data and store it to a file.

Transmitter Block

1. **UDP Source** block takes as input the data packets that are streamed over the UDP pipe by VLC Media Player. The *IP Address* is set to 127.0.0.1 (address of UDP stream) and *Port* set to 1234. *Payload Size* is left at its default value.
2. The **Packet Encoder** block encodes the incoming data packets to symbols using bit sequences for each symbol, giving as output a bit stream. *Samples/Symbol* is set to 2 and *Bits/Symbol* set to 1 (default for GMSK modulation).
3. **GMSK Mod** performs the GMSK modulation using the incoming encoded bit stream as modulating signal. It outputs a GMSK modulated waveform. The modulated waveform is increased in amplitude and then transmitted to USRP using the **UHD: USRP Sink** block. The input and GMSK modulated waveforms are observed using the **QT GUI Sink** block.

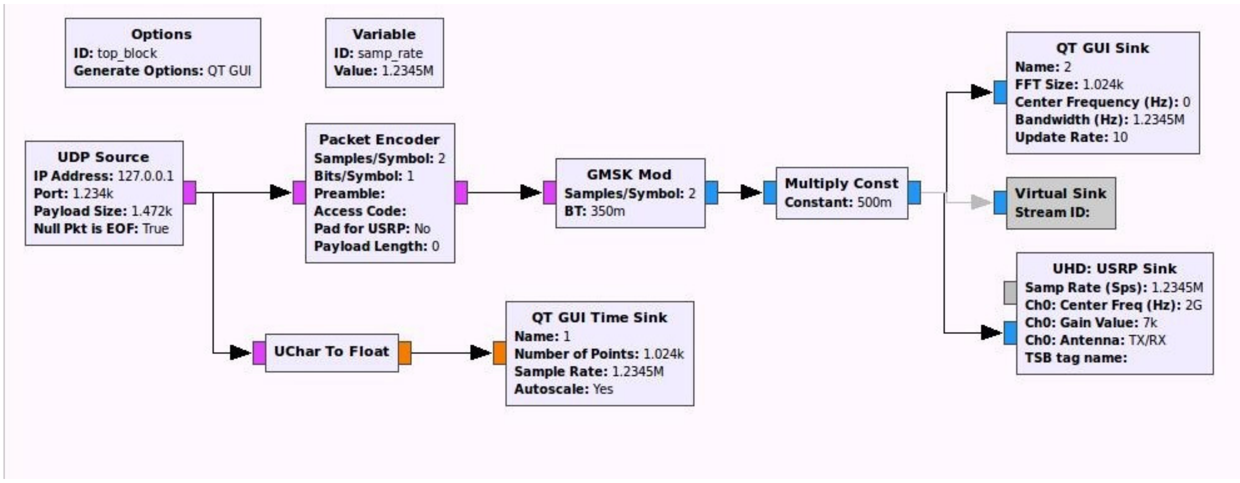


Figure 3 *GNU Radio flow diagram for transmitter station*

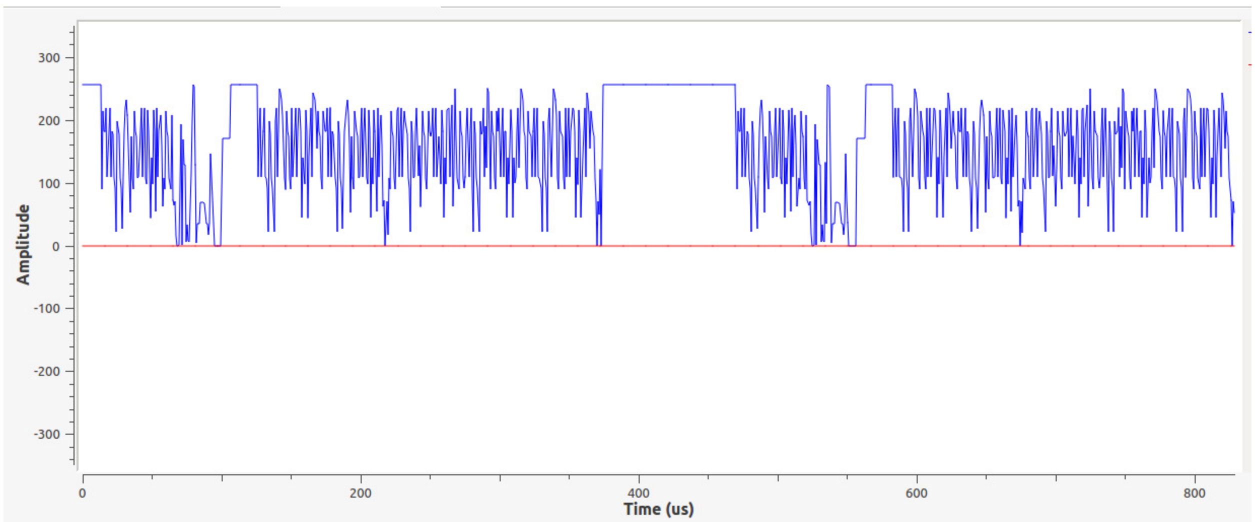


Figure 4 *Input video stream data in time domain*

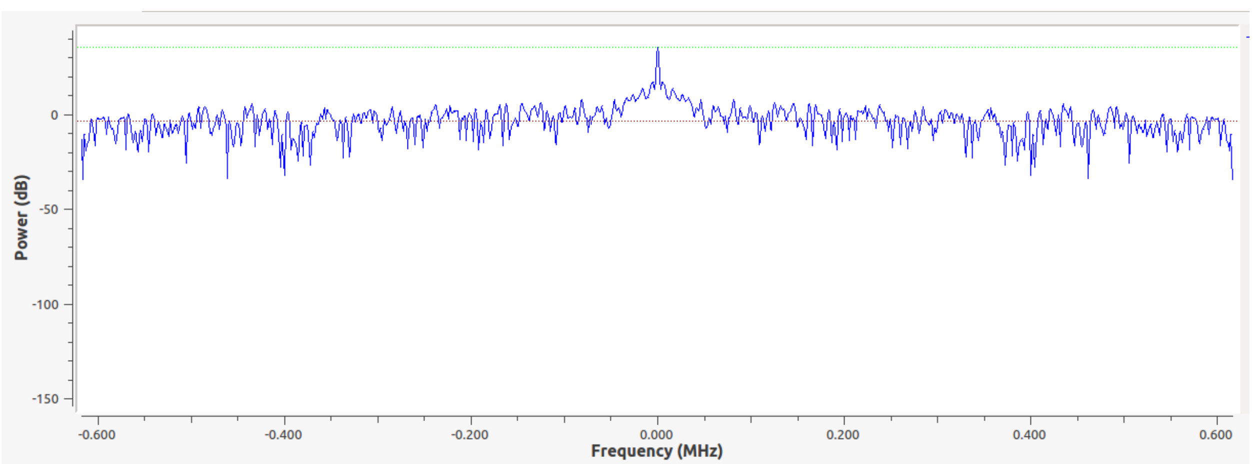


Figure 5 *Input video stream data frequency spectra*

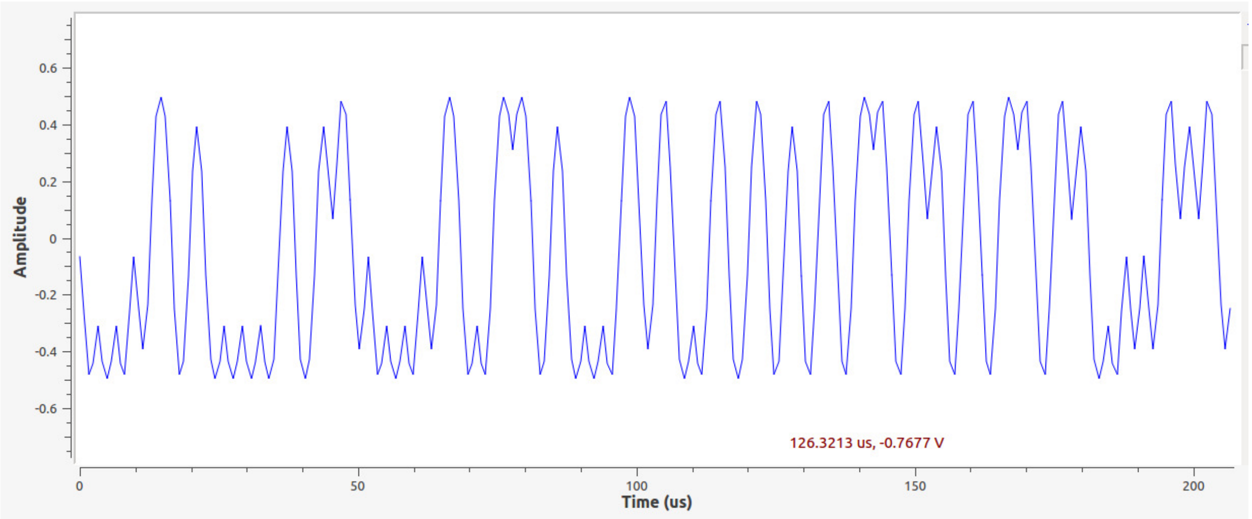


Figure 6 *GMSK modulated wave time response*

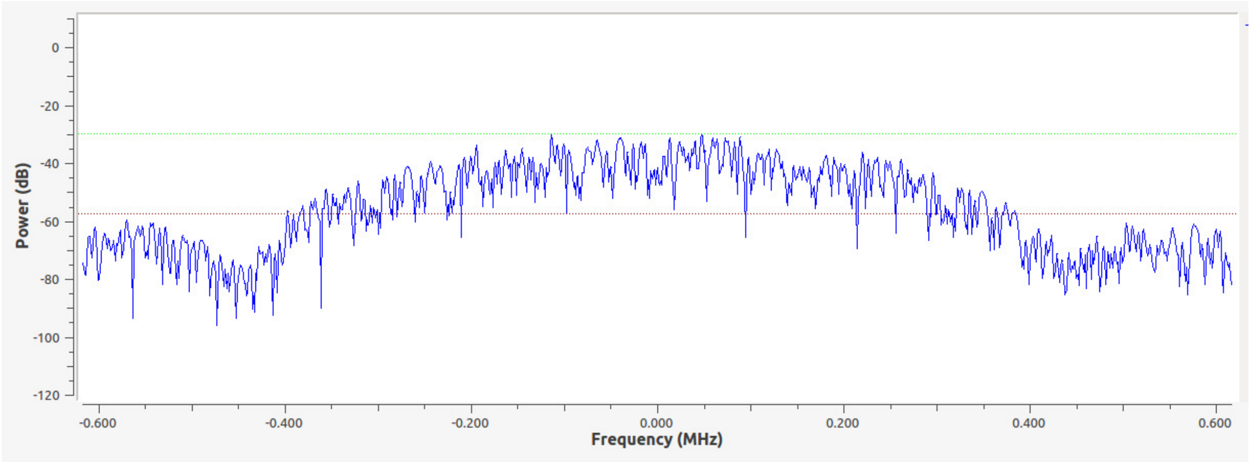


Figure 7 *GMSK modulated wave frequency spectra*

Receiver Block

1. **UHD: USRP Source** takes as input the transmitted waveform of the GMSK modulated wave. A **Low Pass Filter** is used to filter out the high frequency channel noise.
2. **GMSK Demod** demodulates the waveform and outputs a bit stream consisting of the live stream data. This data is decoded by the **Packet Decoder** to data packets. These data packets are appended and stored in a video file (.ts) using the **File Sink**. The stored file may be viewed using a video player or VLC which should display the live video recorded from the input video recording device. The output waveforms are observed using the **QT GUI Sink** block.

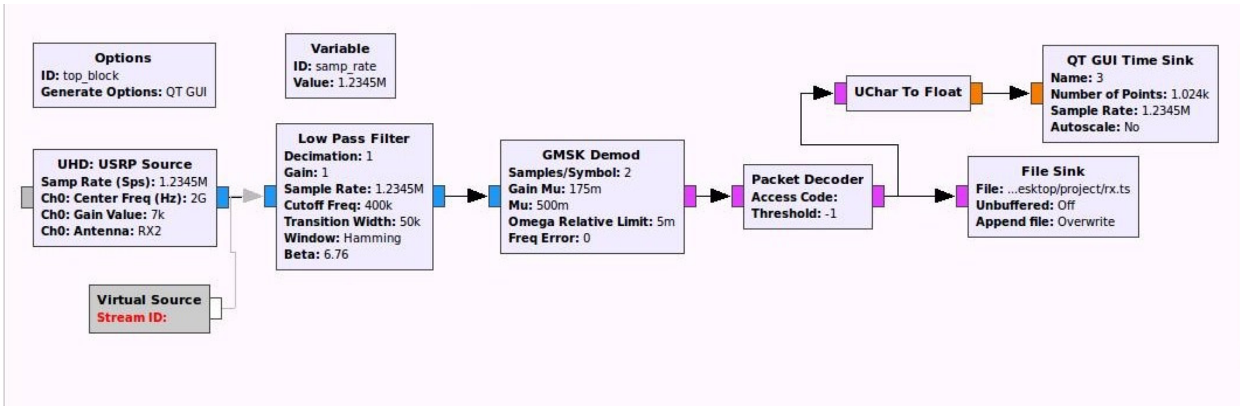


Figure 8 *GNU Radio flow diagram for receiver station*

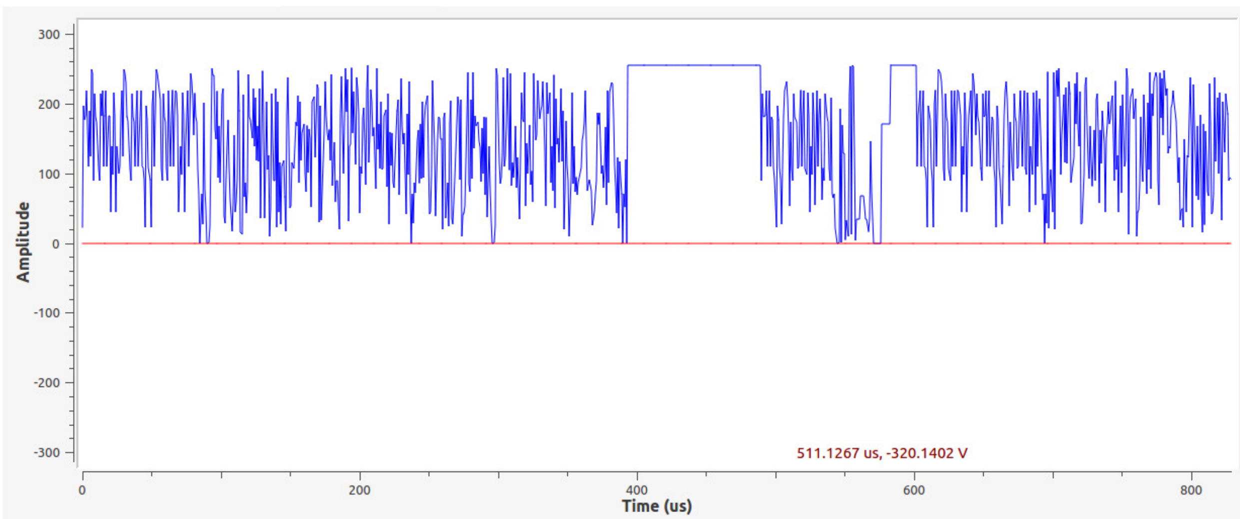


Figure 9 *Output video stream data in time domain*

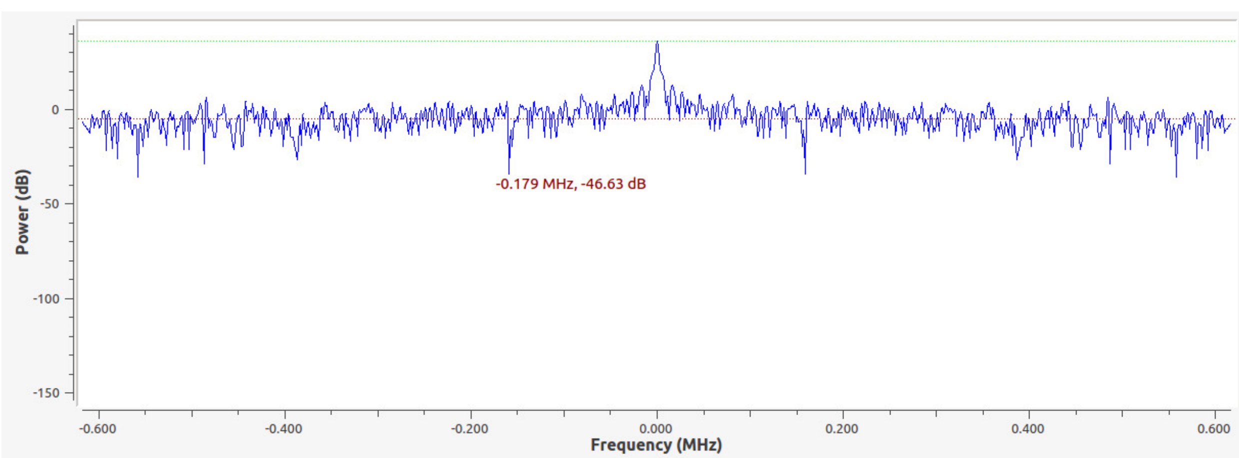


Figure 10 *Output video stream data frequency spectra*

Result and Conclusion

The received waveform was low pass filtered to remove channel noise components. However, some noise still persisted in the received video. Also, a delay of about *5 s* existed between input video recording and output stream. We observed that the quality of video (noise and delay) was highly sensitive to antenna gain variations (of the USRP B200). Except the channel noise and time delay, the received video quality was appreciable, showing that GMSK performs well even in transmission of continuous real-time data.

References

1. *https://wiki.gnuradio.org/index.php/Guided_Tutorial_Extras_Sample_Rates*
2. *http://www.wu.ece.ufl.edu/projects/wirelessVideo/project/H264_USRP/index.html*
3. *http://wiki.oz9aec.net/index.php/Simple_DVB_with_Gstreamer_and_GNU_Radio*