ELEN4012 - Feature Based Automatic Modulation Classification

Jacques Visser and Anthony Farquharson

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

Abstract: Automatic modulation classification involves identifying the modulation scheme used in a signal without the decision being guided by an operator. This report covers a preliminary investigation into the design and implementation of such a system. An overview of the relevant literature is presented and proposals are made regarding the details of the implementation and testing of such a system using and Ettus USRP.

Key words: modulation, classification, USRP, UHD

1. INTRODUCTION

2. LITERATURE SURVEY

Zhu and Nandi [1] identifies three major approaches to automatic modulation classification; likelihood-based, distribution-test-based and feature-based. These are briefly detailed below.

2.1 Likelihood Based Classification

2.2 Distribution Test Based Classification

2.3 Feature Based Classification

Feature based AMR has been shown to be non-ideal, but significantly less computationally intensive [1] than the aforementioned methods.

There are again three major approaches to feature-based AMC. These make use features derived from either the signal spectrum, the wavelet transform of the signal or high-order statistical representations of the signal [1].

The classification of analog modulation schemes using spectral features is well documented by Zhu and Nandi [1] as well as Azzouz and Nandi [2]. They make use of nine features, which are as follows:

1. γ_{max}: Maximum value of Power Spectral Density
2. σ_{ap}: Standard deviation of the absolute value of the non-linear component of the instantaneous phase.
3. σ_{dp}: Standard deviation of the non-linear component of the direct instantaneous phase.
4. P: Spectrum symmetry
5. σ_{aa}: Standard deviation of the absolute value of the normalized and centered instantaneous amplitude
6. σ_{af}: Standard deviation of the absolute value of the normalized and centered instantaneous frequency
7. σ_a: Standard deviation of the normalized and centered instantaneous amplitude
8. μ_{42}^a: Kurtosis of the normalized and centered instantaneous amplitude
9. μ_{42}^f: Kurtosis of the normalized and centered instantaneous frequency

The formulae for the computation of these features can be found in either *Automatic Modulation Classification: Principles, Algorithms and Applications* by Zhu and Nandi [1] or *Automatic modulation recognition of communication signals* by Azzouz and Nandi [2], and are omitted here for brevity.

3. EXISTING SOLUTIONS AND APPLICATIONS OF AMC

3.1 Military

3.2 Civilian

4. SOLUTION SELECTION

5. DESIGN PROCESS OVERVIEW

5.1 Development Methodology

5.2 Estimated Project Schedule

6. IMPLEMENTATION OVERVIEW

6.1 Software

6.1.1 Basic Software Structure Due to the computationally intensive and time-dependent nature of the system, the software would have to be threaded, with each major component running on it's own thread.

The software would consist of various components centered around a main control loop. The USRP Hardware Driver (UHD), run in it's own thread, passes the signal to the central control structure, which filters the signal and passes it to various feature extraction functions, all run in parallel. These functions deliver the features that have been extracted form the signal to a classifier which finally determines the modulation scheme used. This process is represented as a flow diagram in Figure 1. The Feature extraction process is shown in Appendix A and discussed in Section 6.2.

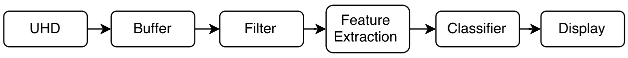

Figure 1 : Overview of the modulation classification process

6.1.2 Libraries and API's Due to the complicated nature of the software implementation, all of the components will not be locally developed. Rather, various API's and libraries will be used.

Firstly, the Ettus USRP Hardware Driver's (UHD) C++ API will be used for communication with the USRP. This is a free & open source piece of software released under the GPLv3 license [3], and is thus free to be used for a project such as this.

In order to spectrally analyze the incoming signal the FFTW3 library is to be used. FFTW3 is an open-source C library developed by Matteo Frigo and Steven G. Johnson. It is available under the GPLv3 license [4].

For displaying graphical information SFML (Simple and Fast Multimedia Library) will be used. This is distributed under the zlib/libpng License and may thus be freely used in a project such as this [5].

For threading and other miscellaneous functions the C++ 11 Standard library will be used. GNU libstdc++ which is used with GCC is distributed under the GPLv3 license.

6.1.3 Build System Due to the open-source nature and wide availability of the libraries and API's used in this project the software produced will be able to run on various platforms. To fully support this the CMake build-system-generator will be used so that this project may easily be compiled on various platforms.

CMake is distributed under the BSD 3-Clause license [6] and it can thus be distributed with and used in this project.

6.2 Feature Extraction

Individual functions will have to be developed for the extraction of each of the nine features mentioned in section 2.3. Some of these features incorporate common data, thus the process of feature extraction may be accelerated by doing the computation thereof only once.

A spectral representation of the signal is used in the process of obtaining most of the features. This is computed with an FFT using the FFTW3 library.

To obtain the instantaneous phase of the signal, the negative portion of the spectral representation of the signal is to be removed and an IFFT is to be performed [7]. Following this, phase unwrapping must be performed [7, 8].

The Instantaneous frequency of the signal may then be obtained from the instantaneous phase by differentiation [8].

This results in a complex series of events. They are shown in sequence as a flow diagram in Appendix A, Figure 1.

6.3 Signal Filtering

Filtering of the input signal to isolate the bandwidth in which a single signal exists is a delicate process. Phase distortion may negatively affect the results as some of the features are dependant on the non-linear portion of the phase only. Thus it is important to implement zero-phase digital filtering. This may be done by either performing forward and reverse filtering or by selecting an FIR filter with a vary flat phase response [9].

The former option would require buffering a large portion of the signal, but might yield better results than the latter. This trade-off, as well as selecting the bandwidth of the band-pass filter to be used is still under investigation.

Filtering will be implemented by convolution as the alternative, multiplication in the Fourier domain, would result in an extra step in feature extraction: finding the magnitude of the resulting signal after performing the IFFT. (See Appendix [**?**] for more information)

6.4 Classifier

6.5 Estimated Costs and Hardware Required

The practical implementation of this project will require at least two USRP devices. The first of which will be used for receiving radio signals, the modulation of which is to be classified. The second will be used to generate modulated signals in order to practically test the operation of the system. Seeing as Wits has these available, no charges would be incurred.

To avoid significant effort in compiling and installation of the USRP Hardware Driver (UHD) two computers running Linux natively will be required. The teammembers' laptops will be used for this purpose.

7. PROPOSED TESTING PROCEDURE

7.1 Simulated Testing

7.2 Practical Testing

8. PRELIMINARY RESULTS

Simple testing of the feature-based AMC algorithm was performed in MATLAB with a limited number of features. The results are promising, showing a clear separation between different modulation schemes. This can be seen in Figure 2.

The marked clustering of the results show that it is indeed possible to make use of a simple threshold-based decision-tree classifier. Making use of such a classifier, however, would require intense interaction with the application to set the appropriate thresholds. Thus it has been deemed appropriate to make use of a K-Nearest-Neighbor clustering algorithm so as to maximise the application's autonomy and extensibility.

9. CONCLUSION AND RECOMMENDATIONS

REFERENCES

[1] Z. Zhu and A. K. Nandi. *Automatic Modulation Classification: Principles, Algorithms and Applications*. John Wiley & Sons, 2015.

[2] E. Azzouz and A. Nandi. *Automatic modulation recognition of communication signals*. Springer Science & Business Media, 2013.

[3] M. Ettus. "Licensing Issues Related to UHD.", 2014. URL `http://lists.ettus.com/pipermail/usrp-users_lists.ettus.com/2014-February/008711.html`.

[4] M. Frigo and S. G. Johnson. "kissfft.", 2013. URL `http://www.fftw.org/doc/License-and-Copyright.html`.

[5] L. Gomila. "SFML License.", 2015. URL `http://www.sfml-dev.org/license.php`.

[6] Kitware. "CMake License.", 2014. URL `http://www.cmake.org/licensing/`.

[7] B. Picinbono. "On instantaneous amplitude and phase of signals." *Signal Processing, IEEE Transactions on*, vol. 45, no. 3, pp. 552–560, 1997.

[8] T. Park. *Introduction to Digital Signal Processing: Computer Musically Speaking*. World Scientific Publishing Company Pte Limited, 2009. URL `https://books.google.co.za/books?id=5LUuUglM5b8C`.

[9] D. Sundararajan. *Digital Signal Processing: Theory and Practice*. World Scientific, 2003. URL `https://books.google.co.za/books?id=Ch5CtUOsAPkC`.

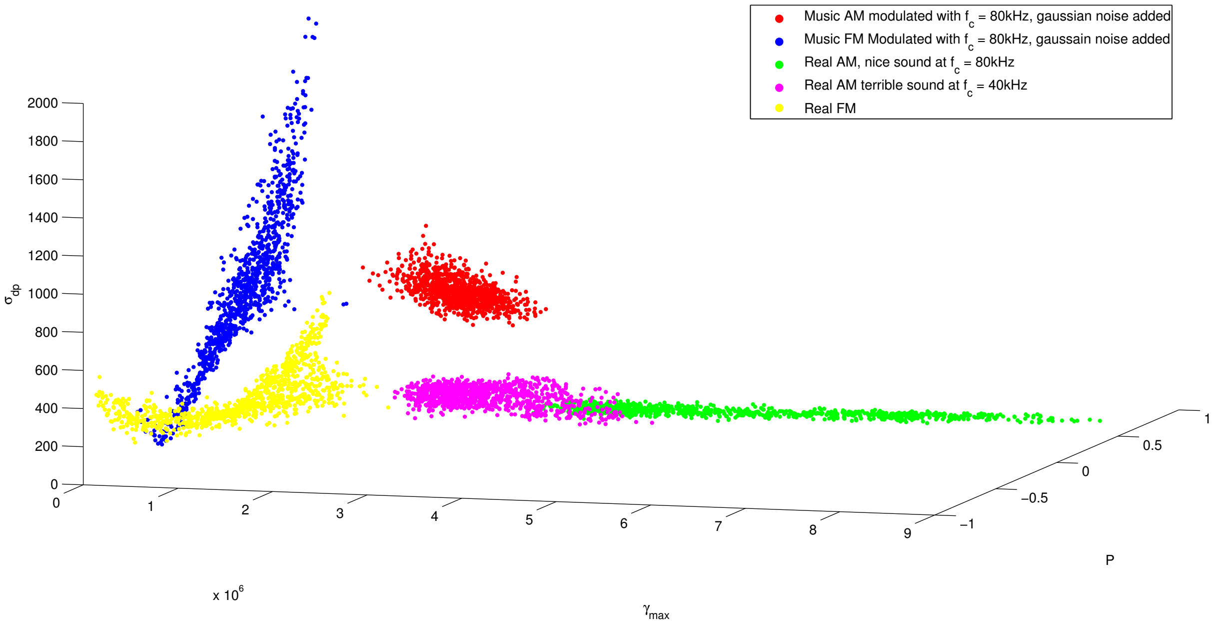

Figure 2 : Plot of σ_{dp} vs. γ_{max} vs. P for various AM and FM signals

Appendix

A Feature Extraction Process

The process of feature extraction is fairly complex. It is presented as a flow graph in Figure. 1 below. Because of the nature of the process it is amenable to parallelization. This will be performed using the C++ 11 Standard threading library.

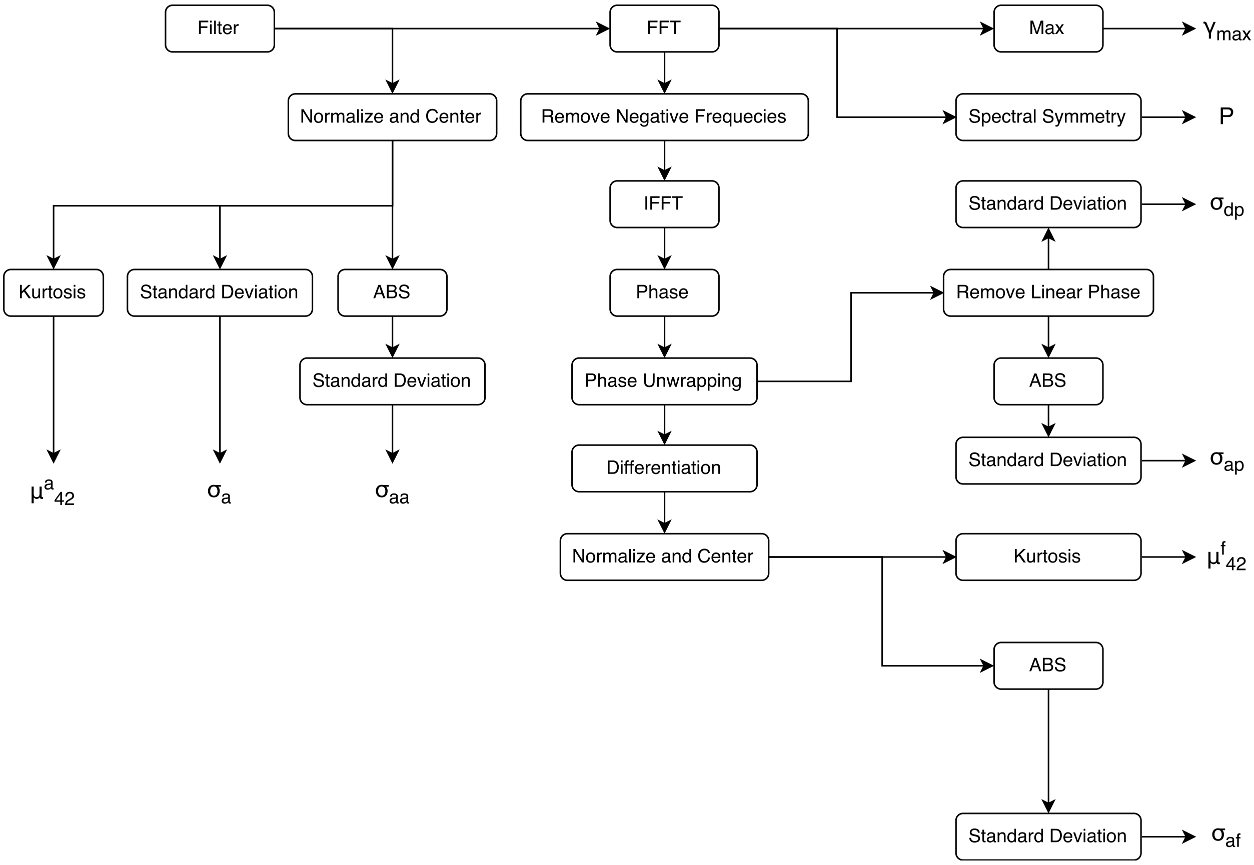

Figure 1 : Feature extraction process as a flow graph

Filtering of the signal prior to feature extraction is done by means of convolution. The alternative, filtering by multiplication in the Fourier domain, would imply that both the phase and magnitude of the output of the IFFT in Figure 1 need to be found. The magnitude found here is equivalent to the input of the FFT. This approach would not only impact the accuracy of the instantaneous amplitude measure, but also introduce another unnecessary operation.