Amplitude Modulation Simulation Software

June 3, 2025

Contents

1 System Overview

The AM simulation software is a software written in Java with user interface (UI) built using Swing.

- **AMSignal**: Represents the AM signal with attributes for time, message, carrier, modulated, demodulated signals, and frequency spectra.
- **ControlPanel**: Provides a GUI for configuring signal parameters (e.g., frequency, amplitude, modulation index) and initiating actions like signal updates or data exports.
- **ModulationAndDemodulation**: Implements core signal processing for modulation, demodulation, noise addition, and spectrum computation.
- **SignalPlotPanel**: Visualizes time and frequency domain plots with zoom and pan capabilities.
- **TimeDomainSimulationWindow**: Supports real-time time-domain signal animation.
- **SpectrumAnalysisFFT**: Performs Fast Fourier Transform (FFT)-based spectrum analysis.
- **TotalHarmonicDistortion**: Analyzes Total Harmonic Distortion (THD) with harmonic visualization.
- **SignalToNoiseRatio**: Computes and displays Signal-to-Noise Ratio (SNR).
- **DataExporter**: Exports signal data to CSV files for external analysis.

The main application (`Main.java`) integrates these components into a tabbed interface, with each tab dedicated to a specific AM variant.

2 Functioning Logic

The software operates through a user-driven workflow:

1. **Parameter Configuration**: Users input signal parameters (e.g., carrier frequency f_c, message frequency f_m, modulation index m) via the `ControlPanel`.
2. **Signal Generation**: The `AMSignal` class generates the message and carrier signals based on user inputs. For a sinusoidal message signal $m(t) = A_m \cos(2\pi f_m t)$ and carrier $c(t) = A_c \cos(2\pi f_c t)$, the modulated signal is computed according to the AM variant.
3. **Modulation**: The `ModulationAndDemodulation` class applies the appropriate modulation technique (e.g., DSB-AM: $s(t) = A_c[1 + m \cdot m(t)] \cos(2\pi f_c t)$).
4. **Demodulation**: Demodulation is performed (e.g., envelope detection for DSB-AM) to recover the message signal.
5. **Visualization**: The `SignalPlotPanel` displays time-domain signals and frequency spectra, while `SpectrumAnalysisFFT` provides dynamic FFT analysis.
6. **Analysis**: THD and SNR are computed and visualized using dedicated modules.
7. **Export**: Users can export signal data to CSV files via `DataExporter`.

3 Simulation and Modeling

The simulation is based on discrete-time signal processing, with signals sampled at a rate f_s (default: 100 kHz) to ensure Nyquist compliance ($f_s \geq 2 \cdot \max(f_c, f_m)$). The `AMSignal` class models signals as arrays of doubles, representing sampled values over a time interval T.

3.1 Time-Domain Simulation

The `TimeDomainSimulationWindow` animates signal propagation in real time, updating the display at a fixed frame rate (e.g., 60 FPS). The simulation loop:

- Generates signal samples for the current time window.
- Updates the modulated and demodulated signals.
- Renders the signals on a canvas using `SignalPlotPanel`.

3.2 Frequency-Domain Modeling

Frequency-domain analysis is performed using FFT, implemented in `SpectrumAnalysisFFT`. The FFT transforms the time-domain signal $x[n]$ into its frequency components $X[k]$:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \quad k = 0, 1, \ldots, N-1$$

where N is the number of samples. The resulting spectrum is visualized with amplitude (dB) versus frequency (Hz).

4 Utilization of Algorithms

The software employs several algorithms for signal processing and analysis:

4.1 Fast Fourier Transform (FFT)

The FFT algorithm (Cooley-Tukey) is used for efficient spectrum computation. The `SpectrumAnalysisFFT` class:

- Samples the signal over a window of N points.
- Applies a Hamming window to reduce spectral leakage.
- Computes the FFT to obtain the frequency spectrum.
- Updates the spectrum display dynamically.

4.2 Modulation Algorithms

The `ModulationAndDemodulation` class implements modulation for each AM variant:

- **DSB-AM**: $s(t) = A_c[1 + m \cdot m(t)] \cos(2\pi f_c t)$.
- **DSB-SC**: $s(t) = A_c \cdot m(t) \cos(2\pi f_c t)$.
- **SSB**: Filters one sideband using a Hilbert transform-based approach.
- **VSB**: Retains a vestige of one sideband using a bandpass filter.
- **QAM**: Combines two modulated signals in quadrature: $s(t) = I(t) \cos(2\pi f_c t) + Q(t) \sin(2\pi f_c t)$.

4.3 Demodulation Algorithms

Demodulation techniques include:

- **Envelope Detection** (DSB-AM): Applies a rectifier and low-pass filter to extract $m(t)$.
- **Coherent Detection** (DSB-SC, SSB): Multiplies the received signal by a synchronized carrier and filters the result.

- **QAM Demodulation**: Uses two coherent detectors for in-phase (I) and quadrature (Q) components.

4.4 Noise Addition

Gaussian noise is added to simulate real-world conditions, with variance controlled by the SNR parameter. The noise is generated using a random number generator and added to the modulated signal.

5 Physics Models

The software is grounded in the physics of electromagnetic wave modulation:

5.1 Signal Representation

Signals are modeled as continuous-time functions sampled discretely. The message signal $m(t)$ is typically a low-frequency sinusoid, while the carrier $c(t)$ is a high-frequency sinusoid. The modulated signal $s(t)$ combines these according to the AM variant.

5.2 Frequency Spectrum

The frequency spectrum of an AM signal is derived from its Fourier transform. For DSB-AM, the spectrum includes:

$$S(f) = \frac{A_c}{2}\left[\delta(f-f_c)+\delta(f+f_c)\right] + \frac{mA_c}{2}\left[M(f-f_c)+M(f+f_c)\right]$$

where $M(f)$ is the message spectrum, and $\delta(f)$ represents the carrier impulses.

5.3 Total Harmonic Distortion (THD)

THD is computed as the ratio of the power of harmonic components to the fundamental frequency:

$$\text{THD} = \sqrt{\frac{\sum_{n=2}^{\infty} P_n}{P_1}}$$

where P_n is the power of the n-th harmonic, extracted via FFT.

5.4 Signal-to-Noise Ratio (SNR)

SNR is calculated as:

$$\text{SNR (dB)} = 10\log_{10}\left(\frac{P_{\text{signal}}}{P_{\text{noise}}}\right)$$

where P_{signal} and P_{noise} are the signal and noise powers, respectively, computed from the FFT spectrum.