



Univerzitet u Sarajevu
Elektrotehnički fakultet
Odsjek za Telekomunikacije

Dizajn digitalnog predajnika i modulacija signala

Telekomunikacijski softver inženjering

Radili:

Vedad Crnčalo, Amna Bumbul, Harun Dedović, Muhamed Crnčalo

12. januar 2025.

Sadržaj

Uvod	2
1 Pojam modulacije	3
2 Amplitudna modulacija (AM) i Amplitude Shift Keying (ASK)	4
2.1 Amplitudna modulacija (AM)	4
2.2 Amplitude Shift Keying (ASK)	5
3 Frekventna modulacija (FM) i frequency shift keying (FSK)	6
3.1 Frekventna modulacija	6
3.2 Frequency shift keying	6
4 PSK (Phase shift keying) i QAM (Quadrature amplitude modulation)	8
4.1 PSK (Phase shift keying)	8
4.1.1 BPSK	8
5 QAM (Quadrature amplitude modulation)	8
6 Digitalne/Analogne modulatorske tehnike - kodovi	9
6.1 Amplitudna modulacija (AM)	9
6.1.1 Osnovna funkcija modulacije	9
6.1.2 Funkcija vizualizacije	9
6.2 Amplitude-Shift Keying (ASK)	10
6.2.1 Osnovna funkcija modulacije	10
6.2.2 Funkcija vizualizacije	10
6.3 Frekvencijska modulacija (FM)	11
6.3.1 FM Modulatorska Funkcija	11
6.3.2 Plotting FM Signala	11
6.3.3 Main FM Modulation Execution	12
6.4 Frequency-Shift Keying (FSK)	13
6.4.1 Osnovna funkcija modulacije	13
6.4.2 Main Execution Funkcija	13
6.5 Binary Phase-Shift Keying (BPSK)	14
6.5.1 Osnovna funkcija modulacije	14
6.5.2 Vizualizacija sa funkcijom uvećanja	15
6.6 QAM Modulatorska Funkcija	15
6.6.1 Plotting QAM Signala	16
6.7 QPSK Modulatorska Funkcija	17
Popis slika	19

Uvod

Savremeni digitalni komunikacijski sistemi čine temelj tehnologija koje omogućavaju brzu i pouzdanu razmjenu informacija. Predajnik, kao ključna komponenta ovih sistema, odgovoran je za generisanje i modulaciju signala prije nego što se on prenese kroz komunikacioni kanal. Ovaj projekat ima za cilj razvijanje osnovnog digitalnog predajnika, što će studentima omogućiti dublje razumijevanje osnovnih principa digitalnih komunikacija, s posebnim fokusom na kreiranje signala, implementaciju raznovrsnih tehnika modulacije i procjenu performansi prenosa.

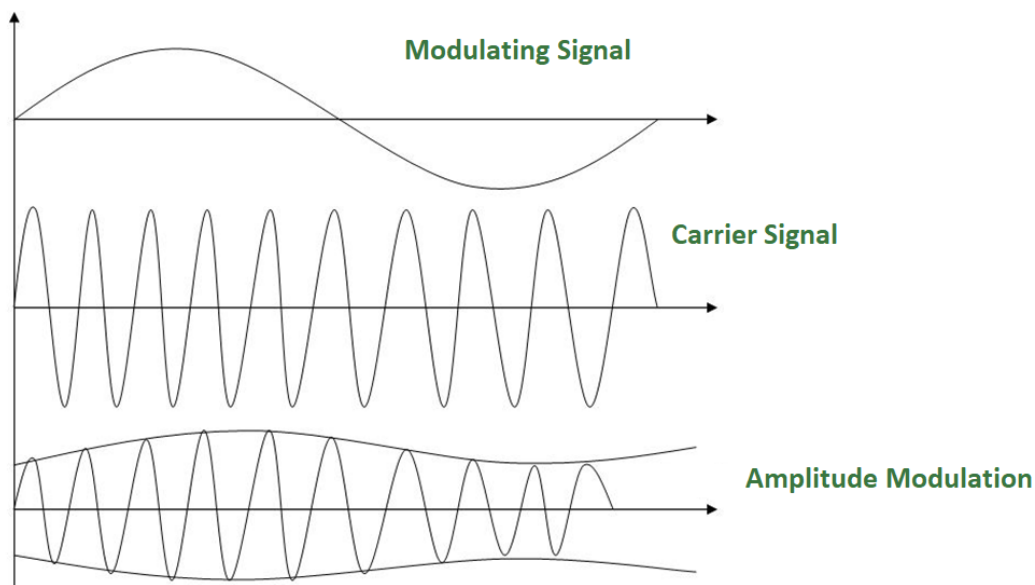
U okviru projekta koristi se programski jezik Python za implementaciju predajnika, dok su među analiziranim šemama modulacije uključene amplitudna modulacija (AM), frekventna modulacija (FM), fazna modulacija (PSK) i kvadratura amplitudna modulacija (QAM). Također, simulacija komunikacionog kanala sa šumom omogućava ispitivanje otpornosti signala u stvarnim uslovima prenosa, pri čemu se kvaliteta signala ocjenjuje analizom odnosa signal-šum (SNR). Na osnovu sprovedenih analiza, studenti će procijeniti efikasnost prenosa za svaku tehniku modulacije, u pogledu energetske efikasnosti i otpornosti na šum.

Cilj ovog projekta je omogućiti praktično iskustvo u dizajniranju i analizi digitalnih komunikacijskih sistema. Kroz ove aktivnosti, razviti će se ključne vještine potrebne za projektovanje, evaluaciju i optimizaciju komunikacijskih tehnologija, čime se unaprijeđuju svoje tehničke i profesionalne sposobnosti u oblasti telekomunikacija.

1 Pojam modulacije

Za prenos informacije između dvije tačke potrebno je informaciju prilagoditi mediju, pretvarajući je u oblik pogodan za prenos. Ovaj proces prilagodbe signala naziva se modulacija, dok se na prijemnoj strani izvodi suprotan proces– demodulacija, koja vraća signal u izvorni oblik.

Modulacija mijenja jedan ili više parametara pomoćnog signala, nazvanog prenosni signal ili nosioc, u skladu sa signalom koji nosi informaciju. Prenosni signal ima veću frekvenciju i bolja svojstva širenja kroz medij. Rezultat modulacije je modulirani signal, dok demodulacija ponovo vraća informaciju.



Slika 1: *Primjer amplitudne modulacije* [4]

Modulacija se obavlja u uređaju zvanom modulator, a demodulacija u demodulator. Kod digitalnog prenosa, uređaji za oba procesa nazivaju se modemima.

Postoje dvije vrste modulacije: prema vrsti modulišućeg signala (kontinuirana ili diskretna) i prema vrsti prenosnog signala. Osnovne vrste analogne modulacije su **amplitudna (AM)**, **frekventna (FM)** i **fazna (PM) modulacija**. Prenosni signali mogu biti sinusni, periodični impulsi ili druge vrste signala. Također postoje i digitalne modulacije kao što su **Amplitude Shift Keying (ASK)**, **Frequency Shift Keying (FSK)**, **Phase Shift Keying (PSK)**.

2 Amplitudna modulacija (AM) i Amplitude Shift Keying (ASK)

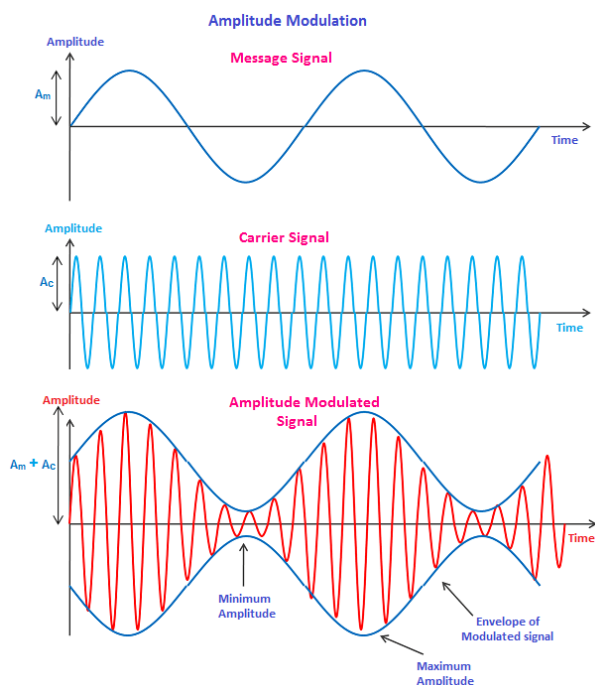
2.1 Amplitudna modulacija (AM)

Amplitudna modulacija je vrsta modulacije u kojoj se amplituda nosećeg signala mijenja u skladu s amplitudom signal poruke, dok faza i frekvencija ostaju konstantni.

U ranim danima bežične komunikacije, komunikacija se obavljala telegrafijom, pri čemu je emitovani signal bio prekinuti radio talas. Kasnije se amplituda ovog talasa mijenjala u skladu s (modulirala prema) govornom porukom (umjesto uključenog/isključenog signala putem telegrafske tipke), a poruka se dobijala iz omotača primljenog signala. Radio talas je nazvan ‘noseći talas’, jer se smatralo da nosi govorne informacije sa sobom. Ovaj proces i signal nazvani su amplitudna modulacija ili skraćeno ‘AM’.

U kontekstu radio komunikacija, pred kraj 20. stoljeća, malo koji modulirani signali su sadržavali značajnu komponentu na frekvenciji ‘nosećeg talasa’. Međutim, iako se noseći talas ne emituje, potreba za takvim signalom na predajniku (gdje se generiše modulirani signal), kao i na prijemniku, ostaje ključna za proces modulacije i demodulacije. Upotreba termina ‘noseći talas’ za opisivanje ovog signala zadržala se do danas.

Za razliku od radio komunikacija, današnje radio emitovanje ipak koristi noseći talas. Emitovanjem ovog nosećeg talasa dizajn demodulatora na prijemniku se značajno pojednostavljuje, što omogućava značajne uštede u troškovima. [1]



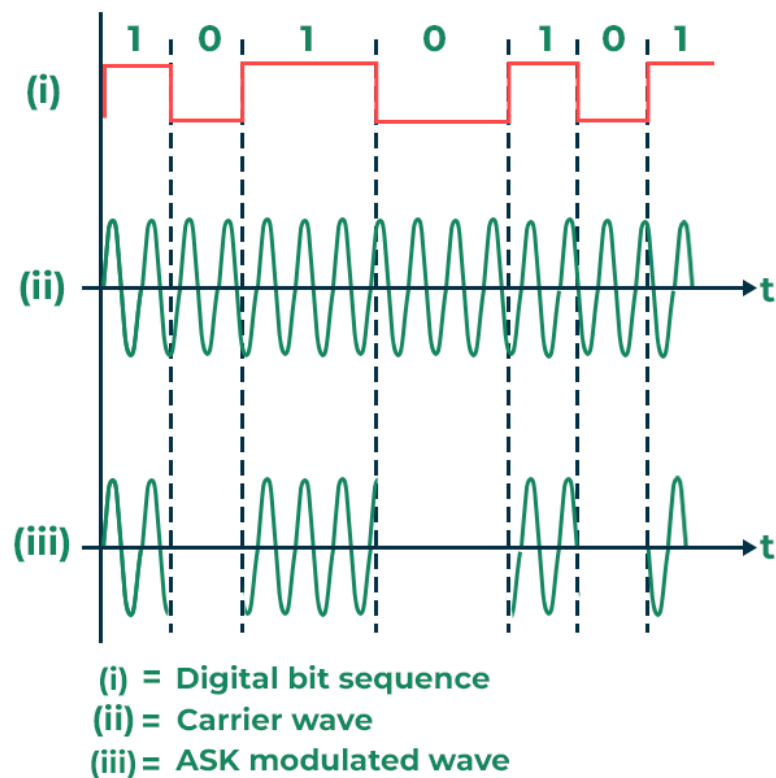
Slika 2: Amplitudna modulacija
[7]

2.2 Amplitude Shift Keying (ASK)

U digitalnoj komunikaciji, različite tehnike modulacije se koriste za prenos podataka ili poruka prijemu preko komunikacionog kanala. Jedna od tih tehnika je (ASK). To je tehnika modulacije koja mijenja amplitudu nosećeg signala kako bi prenijela informacije preko kanala. To je šema modulacije koja ima širok opseg primjene u stvarnom svijetu, uključujući radio, televiziju i digitalni prijenos podataka.

ASK je digitalna tehnika modulacije. Ona prenosi digitalne informacije mijenjanjem amplitude nosećeg signala. U ASK, signal nosioca a s visokom amplitudom koristi se za predstavljanje binarne '1', a signal nosioca s niskom amplitudom predstavlja binarnu '0'.

Ona uključuje superpoziciju nosećeg signala i digitalnog signala poruke. Nosilac signal je često visokofrekventni sinusoidalni talas, koji služi kao nosilac za digitalne informacije. Binarni signal poruke, koji se sastoji od '1' i '0', koristi se za kontrolu amplitude nosećeg signala. Rezultantni signal koji nastaje nakon superpozicije poruke i nosioca prenosi se preko komunikacionog kanala. [4]



Slika 3: *Amplitude Shift Keying*

[4]

3 Frekventna modulacija (FM) i frequency shift keying (FSK)

3.1 Frekventna modulacija

FM je vrsta analogne modulacije koja utiskuje informaciju u frekventnu karakteristiku nosećeg signala. Tako da je frekvencija nosećeg signal izmjenjen prema frekvenciji modulišućeg signala. Sama modulacija je proces modifikovanja signala u oblik pogodan za prijenos informacije kroz medij. Nekoliko zapažanja se može uočiti iz izlaznog FM signala. Amplituda FM signala ostaje konstantna bez obzira na poruku koja se prijenosi, što mu daje svojstvo konstantnog omotača. [2] Faza signala koji se prijenosi također ostaje ne promjenjena. Za razliku od AM-a, FM je otporniji na slabljenje amplitude signala i na šumove koji se pojavljuju tokom prijenosa. Zbog toga je FM privlačniji za korištenje u komercijalnim FM radio sistemima. [3]

Neke od dodatnih primjena su u sistemima za snimanje na magnetnu traku, praćenju novorođenčadi na epileptične napade putem EEG-a, radarima, seizmičkom ispitivanju, sintezi zvuka, telemetriji, dvosmjernim radio-sistemima i sistemima za prijenos videa. [6] Izraz koji opisuje FM modulaciju, to jest modulirani signal je :

$$s(t) = A_c \cos \left(2\pi f_c t + 2\pi k_f \int_0^t m(t) dt \right)$$

gdje su:

- A_c – amplituda nosećeg signala,
- f_c – frekvencija nosećeg signala,
- k_f – osjetljivost na frekvencijsku devijaciju,
- $m(t)$ – modulišući signal.

Na odredišnoj strani prijenosa vrši se demodulacija FM signala, gdje se ekstrahuje informacija iz signala nosioca. Cilj FM demodulacije je reprodukovati originalni signal kodiran u FM signalu, osiguravajući visok kvalitet izlaznog zvuka. Najjednostavnija realizacija FM demodulatora se sastoji od diferencijatora i detektora omotača. Diferencijator vrši konvertovanje FM signala u AM signal, iz kojeg se na detektoru omotača ekstrahuje $m(t)$. [2]

3.2 Frequency shift keying

FSK je modulacija, koja se koristi u digitalnim komunikacijama, u kojoj se svakom simbolu signala koji nosi informaciju dodjeljuje unikatna frekvencija signala nosioca. [5] Digitalni binarni signal mijenja frekvenciju nosioca na dva diskretna nivoa. Ovo omogućava prijemniku da izdvoji digitalni signal demodulacijom. Potrebno je naglasiti da se frekvencija nosioca mijenja u skladu s ulaznim signalom, dok amplituda nosioca ostaje nepromijenjena nakon modulacije. [3] Pored BFSK koji ima dva oblika signala, FSK modulaciju je moguće koristiti kada se želi modulirati grupa od M bita, to jest MFSK modulacije. Na primjer, kod 16-arnog FSK modulacije, uzima se grupa od 4 bita kojima se

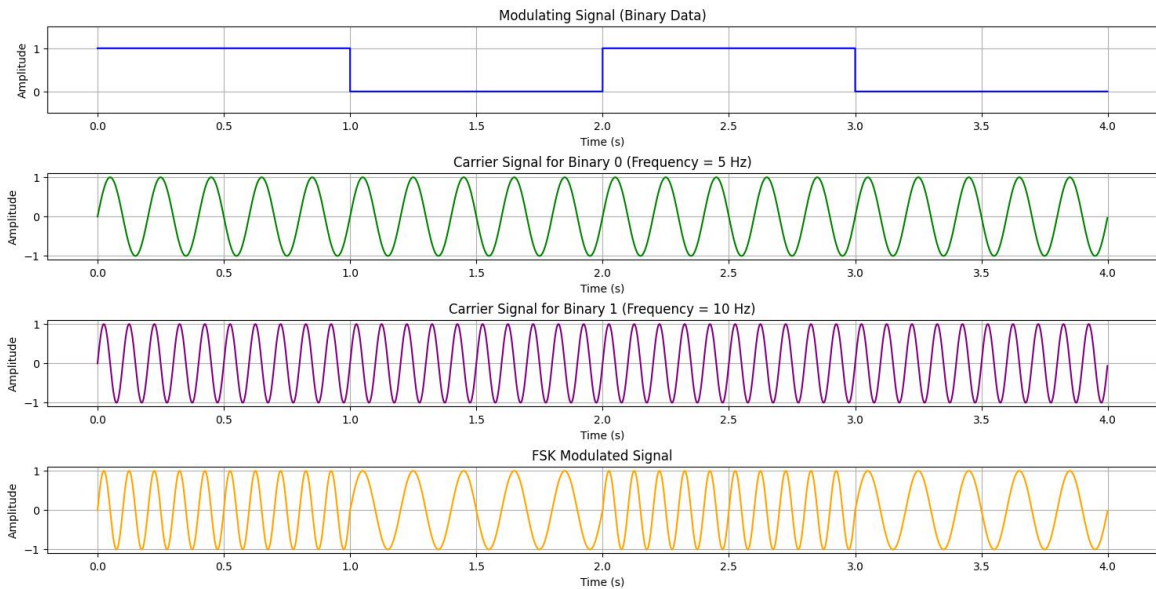
dodjeljuje odgovarajuća frekvencija. Budući da postoji 16 mogućih binarnih kombinacija od 4 bita (2^4), za modulaciju je potrebno 16 različitih frekvencija. Ako se sve te frekvencije generiraju koristeći jednu signal nosioca, takva modulacija se naziva koherentni FSK. S druge strane, ako se za modulaciju koristi više signala nosioca sa različitim frekvencijama, to se naziva nekoherentni FSK. [5] Najjednostavniji izraz (BFSK) koji opisuje FSK je:

$$s(t) = A \cdot \cos(2\pi f_c t + \phi) \cdot m(t)$$

gdje su:

- A – amplituda nosioca,
- f_c – frekvencija nosioca,
- ϕ – početna faza odgovarajućeg nosioca,
- $m(t)$ – modulišući signal.

Nakon što su modulirani binarni podaci preneseni, potrebno ih je primiti i demodulisati. Ovo se često ostvaruje upotrebom band-pass filtera. U slučaju binarne FSK modulacije, prijemnik treba koristiti dva band-pass filtera koji su podešeni na odgovarajuće frekvencije. S obzirom na to da su nominalna nosioca frekvencija i devijacija frekvencije poznate, proces je relativno jednostavan. Jedan band-pass filter biće centriran na frekvenciji Φ_1 , a drugi na Φ_2 . [3]



Slika 4: *Frequency Shift Keying*

4 PSK (Phase shift keying) i QAM (Quadrature amplitude modulation)

4.1 PSK (Phase shift keying)

PSK predstavlja jednu od osnovnih digitalnih modulacionih tehnika. Kod ove vrste modulacije, promjena faze nosećeg signala se koristi da predstavi 0 ili 1 na sljedeći način kod BPSK (Binary phase shift keying-a). [8]

- Prijenos 0 se predstavlja prisutnošću nosioca frekvencije f_c trajanja određenog vremena;
- Prijenos 1 se predstavlja prisutnošću nosioca frekvencije f_c i faznog pomaka u trajanju istog vremena;
- BPSK se matematički može predstaviti na sljedeći način:

$$s(t) = A_c \cos \pi f_c t$$

$$s(t) = A_c \cos 2\pi f_c t + \pi$$

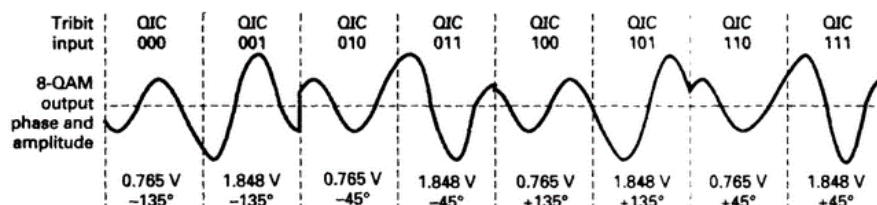
4.1.1 BPSK

BPSK je oblik modulacije kontinualnog (CW) signala. Modulator se ponaša kao prekidač koji obrće fazu, a u zavisnosti od logičkog uslova digitalnog ulaza, nosilac se prenosi do izlaza u fazi ili 180 stepeni van faze sa oscilatorom nosioca reference. U BPSK modulatoru, signal nosilac se množi sa binarnim podacima.

BPSK signalizirajući element (t_s) je jednak vremenu informacijskog bita (t_b), koji ukazuje na to da je brzina prijenosa bita jednaka baud-u (jedinica koja mjeri brzinu prijenosa signala u telekomunikacijama i predstavlja broj promjena ili simbola u sekundi). [8]

5 QAM (Quadrature amplitude modulation)

QAM predstavlja kombinovanu PSK i ASK modulaciju, a koriste se različite veličine ove modulacije. 8-QAM je M-arna tehnika kodiranja gdje je $M = 8$. Za razliku od 8-PSK, izlazni signal iz 8-QAM modulatora nije signal sa konstantnom vrijednosti amplitude. [8]



Slika 5: QAM modulacija
[8]

8-QAM predajnik kao i 8-PSK predajnik radi na sljedeći način: Ulazni podaci se dijele u tri grupe tokova bita (I, Q i C bit). I i Q biti određuju polaritet PAM signala na izlazu konvertera (od drugog do četvrtog stepena), a C kanal određuje opseg. Opsezi I i Q PAM signala su uvijek isti. Njihovi polariteti ovise o logičkom uslovu I i Q bita, i stoga mogu biti različiti. Minimalna vrijednost propusnog opsega je $f_b/3$, kao i kod 8-PSK. 8-QAM prijemnik je gotovo identičan 8-PSK prijemniku. [8]

6 Digitalne/Analogne modulacijske tehnike - kodovi

6.1 Amplitudna modulacija (AM)

6.1.1 Osnovna funkcija modulacije

```
def amplitude_modulation(carrier_freq, sample_rate, duration):
    """
    Perform Amplitude Modulation (AM) of a message signal.

    Parameters:
    - carrier_freq: Frequency of the carrier signal in Hz
    - sample_rate: Sampling rate in Hz
    - duration: Duration of the signal in seconds
    """
    time = np.arange(0, duration, 1 / sample_rate)
    message_signal = np.sin(2 * np.pi * time)
    carrier_signal = np.sin(2 * np.pi * carrier_freq * time)
    am_signal = message_signal * np.cos(2 * np.pi * carrier_freq
        * time)
    return time, message_signal, carrier_signal, am_signal
```

Listing 1: AM Modulation Implementation

6.1.2 Funkcija vizualizacije

```
def plot_am_signals(time, message_signal, carrier_signal,
    am_signal, carrier_freq):
    """
    Plot the message signal, carrier signal, and AM modulated
    signal.
    """
    plt.figure(figsize=(12, 8))

    # Message signal plot
    plt.subplot(3, 1, 1)
    plt.plot(time, message_signal)
    plt.title(r'Message Signal:  $\sin(2\pi t)$ ')
    plt.xlabel('Time (s)')
    plt.ylabel('Amplitude')

    # Carrier signal plot
    plt.subplot(3, 1, 2)
```

```
plt.plot(time, carrier_signal)
plt.title(r'Carrier Signal:  $\sin(2\pi f_c t)$ ,  $f_c =$ ' + f
        '{carrier_freq} Hz')

# Modulated signal plot
plt.subplot(3, 1, 3)
plt.plot(time, am_signal)
plt.title('Amplitude Modulated Signal')
```

Listing 2: AM Signal Visualization

6.2 Amplitude-Shift Keying (ASK)

6.2.1 Osnovna funkcija modulacije

```
def ask_modulation(binary_sequence):
    """
    Generates ASK modulation components for a binary sequence.

    Parameters:
    - binary_sequence: A binary sequence (e.g., [1, 0, 1, 0])
    """
    n = len(binary_sequence)
    bw = np.repeat(binary_sequence, 100) # Expand binary
        sequence
    t = np.linspace(0, n, len(bw))
    sint = np.sin(2 * np.pi * t) # Carrier signal
    st = bw * sint # ASK modulated signal
    return t, bw, sint, st
```

Listing 3: ASK Modulation Implementation

6.2.2 Funkcija vizualizacije

```
def plot_ask_signals(t, bw, sint, st):
    """
    Plots the digital signal, carrier signal, and ASK modulated
        signal.
    """
    plt.figure(figsize=(10, 5))

    # Digital signal plot
    plt.subplot(3, 1, 1)
    plt.plot(t, bw, linewidth=1.5)
    plt.title('Digital Signal')

    # Carrier signal plot
    plt.subplot(3, 1, 2)
    plt.plot(t, sint, linewidth=1.5)
    plt.title('Carrier Signal')
```

```

# ASK modulated signal plot
plt.subplot(3, 1, 3)
plt.plot(t, st, linewidth=1.5)
plt.title('ASK Modulated Signal')

```

Listing 4: ASK Signal Visualization

6.3 Frekvencijska modulacija (FM)

6.3.1 FM Modulacijska Funkcija

```

def fm_modulation(carrier_freq, sample_rate, duration,
freq_deviation):
    """!
    @brief Perform Frequency Modulation (FM) of a message signal.

    @param carrier_freq Carrier frequency in Hz.
    @param sample_rate Sampling rate for signal generation in Hz.
    @param duration Duration of the signal in seconds.
    @param freq_deviation Frequency deviation (maximum shift from
        carrier frequency in Hz).

    @return A tuple containing:
        - time: Time vector for the entire signal.
        - message_signal: The base message signal (a simple sine
            wave).
        - carrier_signal: The carrier signal before modulation.
        - fm_signal: The resulting FM modulated signal.

    @code
    carrier_freq = 10
    sample_rate = 1000
    duration = 2
    freq_deviation = 5
    time, message_signal, carrier_signal, fm_signal =
        fm_modulation(carrier_freq, sample_rate, duration,
            freq_deviation)
    @endcode
    """

```

Listing 5: Frequency Modulation (FM) Function

6.3.2 Plotting FM Signala

```

def plot_fm_signals(time, message_signal, carrier_signal,
    fm_signal, carrier_freq, freq_deviation):
    """!
    @brief Plot the message signal, carrier signal, and FM modulated
        signal.

```

```

@param time Time vector for the signals.
@param message_signal The base message signal.
@param carrier_signal The carrier signal before modulation.
@param fm_signal The resulting FM modulated signal.
@param carrier_freq Carrier frequency in Hz.
@param freq_deviation Frequency deviation in Hz.

@code
plot_fm_signals(time, message_signal, carrier_signal, fm_signal,
    carrier_freq=10, freq_deviation=5)
@endcode
"""

plt.subplot(3, 1, 1)
plt.plot(time, message_signal)
plt.title('Message Signal:  $\sin(2\pi t)$ ')

plt.subplot(3, 1, 2)
plt.plot(time, carrier_signal)
plt.title(f'Carrier Signal:  $\sin(2\pi F_c t)$ ,  $F_c = \{\text{carrier\_freq}\}$  Hz')

plt.subplot(3, 1, 3)
plt.plot(time, fm_signal)

plt.tight_layout()
plt.show()

```

Listing 6: Plot FM Signals

6.3.3 Main FM Modulation Execution

```

def main_fm():
    """
    @brief Main function to execute FM modulation and visualize
        the results.

    Executes the FM modulation process and plots the message,
        carrier, and modulated signals.

    @code
    This function automatically executes when the script runs and
        plots the signals.
    Parameters used: carrier_freq=10, sample_rate=1000, duration
        =2, freq_deviation=5.
    @endcode
    """
    # Define FM modulation parameters.
    carrier_freq = 10          # Carrier frequency (Hz).
    sample_rate = 1000         # Sampling rate (Hz).

```

```

duration = 2          # Duration of the signal (seconds).
freq_deviation = 5    # Frequency deviation (Hz).

# Perform FM modulation.
time, message_signal, carrier_signal, fm_signal =
    fm_modulation(
        carrier_freq, sample_rate, duration, freq_deviation)

# Plot the modulated signals.
plot_fm_signals(time, message_signal, carrier_signal,
    fm_signal, carrier_freq, freq_deviation)

if __name__ == "__main__":
    main_fm() # Run the main FM modulation function.

```

Listing 7: FM Modulation Main Function

6.4 Frequency-Shift Keying (FSK)

6.4.1 Osnovna funkcija modulacije

```

def fsk_modulation(binary_data, carrier_freq_0, carrier_freq_1,
    sample_rate, bit_duration):
    """
    Perform FSK modulation.

    Parameters:
    - binary_data: Binary sequence to modulate
    - carrier_freq_0: Frequency for binary 0
    - carrier_freq_1: Frequency for binary 1
    - sample_rate: Sampling rate in Hz
    - bit_duration: Duration of each bit
    """
    time = np.arange(0, len(binary_data) * bit_duration, 1 /
        sample_rate)
    signal_length = len(time)

    # Initialize signals
    modulating_signal = np.zeros(signal_length)
    fsk_signal = np.zeros(signal_length)

```

Listing 8: FSK Modulation Implementation

6.4.2 Main Execution Funkcija

```

def main_fsk():
    """
    Main function to execute FSK modulation with visualization
    """
    # Default parameters

```

```

carrier_freq_0 = 5      # Frequency for binary 0 (Hz)
carrier_freq_1 = 10     # Frequency for binary 1 (Hz)
sample_rate = 1000      # Sampling rate (Hz)
bit_duration = 1        # Duration of each bit (seconds)

# Get binary sequence from user
binary_data = input("Enter binary sequence (e.g., 1, 0, 1, 0) : ")
binary_data = list(map(int, binary_data.strip().split(',')))

# Perform modulation and plot
time, modulating_signal, carrier_0, carrier_1, fsk_signal =
    fsk_modulation(
        binary_data, carrier_freq_0, carrier_freq_1, sample_rate,
        bit_duration)
plot_fsk_signals(time, modulating_signal, carrier_0,
    carrier_1,
        fsk_signal, carrier_freq_0, carrier_freq_1)

```

Listing 9: FSK Main Function

6.5 Binary Phase-Shift Keying (BPSK)

6.5.1 Osnovna funkcija modulacije

```

def bpsk_modulation(binary_data, carrier_freq, sample_rate,
    bit_duration):
    """
    Perform Binary Phase-Shift Keying modulation.
    """
    time = np.arange(0, len(binary_data) * bit_duration, 1 /
        sample_rate)
    samples_per_bit = int(sample_rate * bit_duration)
    bit_time = np.linspace(0, bit_duration, samples_per_bit,
        endpoint=False)

    # Initialize signals
    modulating_signal = np.zeros(len(time))
    bpsk_signal = np.zeros(len(time))
    carrier_signal = np.sin(2 * np.pi * carrier_freq * time)

    # Generate BPSK signal
    for i, bit in enumerate(binary_data):
        start_idx = i * samples_per_bit
        end_idx = (i + 1) * samples_per_bit

        if bit == 0:
            bpsk_signal[start_idx:end_idx] = np.sin(2 * np.pi *
                carrier_freq * bit_time)
        else:

```

```

        bpsk_signal[start_idx:end_idx] = np.sin(2 * np.pi *
            carrier_freq * bit_time + np.pi)

    return time, modulating_signal, carrier_signal, bpsk_signal

```

Listing 10: BPSK Modulation Implementation

6.5.2 Vizualizacija sa funkcijom uvećanja

```

def plot_bpsk_signals(time, modulating_signal, carrier_signal,
    bpsk_signal, carrier_freq):
    """
    Plot BPSK signals with a zoomed view of the modulated signal.
    """
    plt.figure(figsize=(12, 8))

    # Standard signal plots
    plt.subplot(4, 1, 1)
    plt.step(time, modulating_signal, where='post')
    plt.title('Modulating Signal (Binary Data)')

    plt.subplot(4, 1, 2)
    plt.plot(time, carrier_signal)
    plt.title(f'Carrier Signal (Frequency = {carrier_freq} Hz)')

    plt.subplot(4, 1, 3)
    plt.plot(time, bpsk_signal)
    plt.title('BPSK Modulated Signal')

    # Zoomed view of modulated signal
    plt.subplot(4, 1, 4)
    zoom_start = 0.5
    zoom_end = 1.5
    zoom_start_index = np.argmin(np.abs(time - zoom_start))
    zoom_end_index = np.argmin(np.abs(time - zoom_end))
    plt.plot(time[zoom_start_index:zoom_end_index],
        bpsk_signal[zoom_start_index:zoom_end_index])
    plt.title('Zoomed BPSK Modulated Signal')

```

Listing 11: BPSK Signal Visualization with Zoom

6.6 QAM Modulacijska Funkcija

```

def qam_modulation(binary_data, carrier_freq, sample_rate,
    bit_duration, constellation_points):
    """
    Perform Quadrature Amplitude Modulation (QAM) on a binary
    data sequence.

    Parameters:

```



```

- binary_data: Binary sequence to modulate (e.g., [0, 1, 1, 0]).
- carrier_freq: Carrier frequency in Hz.
- sample_rate: Sampling rate in Hz.
- bit_duration: Duration of each bit in seconds.
- constellation_points: Complex numbers representing constellation points.

Returns:
- time: Time vector for the entire signal duration.
- modulating_signal: Sequence of constellation indices over time.
- carrier_signal_i: In-phase carrier signal.
- carrier_signal_q: Quadrature carrier signal.
- qam_signal: Resulting QAM modulated signal.
"""
# Calculate derived parameters.
bits_per_symbol = int(np.log2(len(constellation_points))) # Bits per symbol.
samples_per_symbol = int(sample_rate * bit_duration) # Samples per symbol.
time = np.arange(0, len(binary_data) / bits_per_symbol * bit_duration, 1 / sample_rate)

# Create carriers.
carrier_signal_i = np.cos(2 * np.pi * carrier_freq * time) # In-phase carrier.
carrier_signal_q = np.sin(2 * np.pi * carrier_freq * time) # Quadrature carrier.

```

Listing 12: Quadrature Amplitude Modulation (QAM) Function

6.6.1 Plotting QAM Signals

```

def plot_qam_signals(time, modulating_signal, carrier_signal_i,
                    carrier_signal_q, qam_signal, carrier_freq):
    """
    Plot the modulating signal, carrier signals, and QAM
    modulated signal.

    Parameters:
    - time: Time vector for the signals.
    - modulating_signal: Constellation indices over time.
    - carrier_signal_i: In-phase carrier signal.
    - carrier_signal_q: Quadrature carrier signal.
    - qam_signal: QAM modulated signal.
    - carrier_freq: Frequency of the carrier signal in Hz.
    """
    plt.figure(figsize=(12, 10))

    # Plot the modulating signal (constellation indices).

```

```

plt.subplot(4, 1, 1)
plt.step(time, modulating_signal.real, label='In-phase
Component')
plt.step(time, modulating_signal.imag, label='Quadrature
Component', linestyle='dashed')
plt.grid(True)
plt.title('Modulating Signal (Constellation Indices)')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.legend()

plt.tight_layout()
plt.show()

```

Listing 13: Plot QAM Signals

6.7 QPSK Modulacijska Funkcija

```

import numpy as np

def qpsk_modulation(binary_data, carrier_freq, sample_rate,
    bit_duration):
    """
    Perform Quadrature Phase Shift Keying (QPSK) modulation on a
    binary data sequence.

    Parameters:
    - binary_data: Binary sequence to modulate (e.g., [0, 1, 1,
      0]).
    - carrier_freq: Carrier frequency in Hz.
    - sample_rate: Sampling rate in Hz.
    - bit_duration: Duration of each bit in seconds.

    Returns:
    - time: Time vector for the entire signal duration.
    - modulating_signal: Sequence of constellation indices over
      time.
    - carrier_signal_i: In-phase carrier signal.
    - carrier_signal_q: Quadrature carrier signal.
    - qpsk_signal: Resulting QPSK modulated signal.
    """
    constellation_points = [1+0j, 0+1j, -1+0j, 0-1j] # QPSK
    constellation
    bits_per_symbol = 2 # QPSK uses 2 bits per symbol.

    # Ensure binary data length is divisible by 2.
    if len(binary_data) % bits_per_symbol != 0:
        raise ValueError("Length of binary data must be a
            multiple of 2 for QPSK.")

    # Derived parameters.

```

```

samples_per_symbol = int(sample_rate * bit_duration)
time = np.arange(0, len(binary_data) / bits_per_symbol *
                 bit_duration, 1 / sample_rate)

```

Listing 14: Quadrature Phase Shift Keying (QPSK) Function

Plotting QPSK Signals

```

import matplotlib.pyplot as plt

def plot_qpsk_signals(time, modulating_signal, carrier_signal_i,
                      carrier_signal_q, qpsk_signal, carrier_freq):
    """
    Plot the modulating signal, carrier signals, and QPSK
    modulated signal.

    Parameters:
    - time: Time vector for the signals.
    - modulating_signal: Constellation points over time.
    - carrier_signal_i: In-phase carrier signal.
    - carrier_signal_q: Quadrature carrier signal.
    - qpsk_signal: QPSK modulated signal.
    - carrier_freq: Frequency of the carrier signal in Hz.
    """
    plt.figure(figsize=(12, 10))

    # Plot the modulating signal.
    plt.subplot(5, 1, 1)
    plt.step(time[:len(time)//len(modulating_signal)],
             modulating_signal.real, label='In-phase')
    plt.step(time[:len(time)//len(modulating_signal)],
             modulating_signal.imag, label='Quadrature', linestyle='
             dashed')
    plt.grid(True)
    plt.title('Modulating Signal (Constellation Points)')
    plt.xlabel('Time (s)')
    plt.ylabel('Amplitude')
    plt.legend()

    plt.tight_layout()
    plt.show()

```

Listing 15: Plot QPSK Signals

Spisak slika

1	<i>Primjer amplitudne modulacije [4]</i>	3
2	<i>Amplitudna modulacija</i>	4
3	<i>Amplitude Shift Keying</i>	5
4	<i>Frequency Shift Keying</i>	7
5	<i>QAM modulacija</i>	8

Literatura

- [1] Auburn University. Analog modulation/manuals/studenttext-vol1, //. Accessed: 2025-01-03.
- [2] Lawrence Der. Frequency modulation (fm) tutorial. *Silicon Laboratories Inc*, 2008.
- [3] Saleh Faruque. *Radio frequency modulation made easy*. Springer, 2017.
- [4] GeeksforGeeks. Amplitude shift keying, 2024. Accessed: 2025-01-03.
- [5] GeeksforGeeks. Digital modulation techniques - frequency shift keying, 2025. Accessed: 2025-01-03.
- [6] GeeksforGeeks. Frequency modulation, 2025. Accessed: 2025-01-02.
- [7] Physics and Radio-Electronics. Amplitude modulation, 2025. Accessed: 2025-01-03.
- [8] Malaysia UniversitiSainsMalaysia, Public university in George Town. Chapter 2 digital modulation 2.1 introduction. Accessed: 2025-01-03.