Name-Soumyadeep Singh

Class-BCSE 3

Roll-002210501018

Section-A1

Computer Networks Lab

Assignment 4

Implement CDMA with Walsh code.

The **Simple_CDMA Project** is designed to simulate a CDMA-based (Code Division Multiple Access) communication system, using Welsh (Hadamard) codes to allow multiple stations to transmit over a single shared channel simultaneously without interference. The simulation reflects the fundamental principles of CDMA, enabling users to understand the mechanisms of code-based multiplexing and signal detection in a shared communication medium. Below is an expanded overview of the major design components:

1. Welsh Code Generation

- **Purpose**: Welsh codes, also known as Hadamard codes, are a type of orthogonal code crucial in CDMA systems. They allow each station to be assigned a unique, orthogonal sequence, which enables simultaneous transmissions without signal overlap or interference on the shared channel. The orthogonality of these codes is essential to CDMA as it ensures that each station's signal can be distinctly separated from others, regardless of their transmission timing or content.
- **Method**: To generate Welsh codes, the project uses a recursive approach within the `welsh_code_generator` function. This function constructs a Hadamard matrix, a mathematical structure that provides the required orthogonal codes for any number of stations, as long as the number is a power of two. By expanding the matrix recursively, it's possible to achieve larger sets of orthogonal codes that maintain the zero cross-correlation properties critical for signal clarity in CDMA.
- **Scalability**: One of the advantages of this design is its scalability. The function calculates the nearest power of two based on the number of stations, dynamically generating codes as needed. This approach makes it adaptable, allowing for flexible adjustments in the number of participating stations, which is especially useful in real-world scenarios where the number of active transmitters may vary.

2. State Simulation of Stations

- **Purpose**: In a real CDMA environment, stations operate in various states depending on whether they are actively transmitting data, sending silence, or idling. In this simulation, each station can be in one of three states: sending a "1" (represented as +1), sending a "0" (represented as -1), or sending nothing (represented as 0). This state diversity is essential for realism, as it reflects the behavior of real CDMA networks where stations may be silent or transmitting different data simultaneously.

- **Randomized Transmission**: To emulate a more realistic CDMA environment, the state of each station is randomized in every iteration of the simulation. This randomness allows each station to alternate between active transmission and silence, mimicking a real-world CDMA system where not all stations are constantly transmitting. This approach provides insights into how a CDMA network handles variable traffic and the robustness of Welsh codes in separating signals regardless of changes in transmission states.

3. Channel Content Calculation

- **Purpose**: In a CDMA system, all station transmissions are combined onto a shared communication channel. This design step simulates how individual station signals are merged into a single channel content vector. Each active station's contribution to the channel is calculated for each bit position, forming a cumulative signal that represents the channel's overall content at any moment.
- **Code Design**: This is achieved through a simple yet effective summation loop, which aggregates the contributions of each station based on its Welsh code and current transmission state. The loop iterates over all stations, calculating the product of each station's state and its assigned Welsh code. The resulting values are then summed to form the channel content vector. This vector reflects the combined signal on the shared channel, allowing each station to interpret and decode its portion of the transmission accurately.

4. Station Signal Decoding

- **Purpose**: Once the channel content vector is generated, each station needs to decode its signal to determine whether it transmitted a "1," a "0," or no signal at all. This decoding process is essential in CDMA as it allows each station to filter out the channel noise and accurately read its own transmission, despite the simultaneous signals from other stations.
- **Signal Detection**: Each station reads from the channel by performing a dot product between the channel content vector and its unique Welsh code. This calculation filters out the other stations' signals due to the orthogonality of the Welsh codes. By normalizing the result, the station can interpret the outcome as either a "1," a "0," or no signal. This precise signal detection demonstrates the strength of orthogonal codes in CDMA, where

each station can reliably decode its signal even amidst concurrent transmissions.

5. Iterative User Interaction

- **Purpose**: This CDMA simulation includes an interactive aspect where users can view each station's state, the calculated channel content, and the decoded values at every iteration. This iterative display allows users to observe CDMA principles in action, making the simulation a valuable educational tool for visualizing the impact of orthogonal codes on simultaneous transmissions.
- **Exit Option**: The simulation is flexible, giving users the choice to exit or continue with additional iterations. This feature is particularly useful for testing purposes, as users can adjust the number of stations, states, and view multiple transmission rounds. This iterative interaction makes the CDMA process more tangible and offers opportunities for further exploration and testing in simulated scenarios.

Overall Design Benefits and Future Enhancements

This modular project design, with separate components for Welsh code generation, station state simulation, channel content calculation, and signal decoding, results in a clear and maintainable code structure. The modularity allows for easy adjustments and scalability, enabling users to modify parameters like the number of stations or transmission probabilities without extensive reconfiguration. Furthermore, the structure can be readily extended to include more advanced features, such as handling complex multi-state transmissions or implementing error-checking mechanisms. Future versions could also include more sophisticated state control for stations, adaptive error correction, and realistic network congestion modeling to create an even more comprehensive CDMA simulation environment.

By providing a flexible and detailed simulation, this project offers valuable insights into the workings of CDMA, demonstrating how Welsh codes ensure clear signal separation on shared channels and showcasing CDMA’s adaptability to varied transmission conditions.

Code

```
import random

import numpy as np

def generate_walsh_codes(order):

    """Generate Walsh codes for a given order using recursive construction."""

    if order == 1:

        return np.array([[1]])

    previous_order = generate_walsh_codes(order // 2)

    top = np.hstack((previous_order, previous_order))

    bottom = np.hstack((previous_order, -previous_order))

    return np.vstack((top, bottom))



def walsh_code_matrix(num_stations):

    """Generate a Walsh code matrix with rows equal to the number of stations."""

    order = 1

    while order < num_stations:

        order *= 2

    walsh_codes = generate_walsh_codes(order)

    return walsh_codes[:num_stations, :]
```

```
def simulate_transmission(num_stations, codes):
    """Simulate transmission by randomly choosing send values and
decoding from channel."""
    bit_length = len(codes[0])
    transmissions = []

    for i in range(num_stations):
        send_value = random.choice([1, -1, 0])
        transmissions.append(send_value)

        transmission_str = "1" if send_value == 1 else "0" if
send_value == -1 else "nothing"
        print(f"Station {i+1} is sending: {transmission_str}")

    channel_signal = np.zeros(bit_length)
    for i in range(num_stations):
        channel_signal += transmissions[i] * codes[i]

    print(f"Channel content: {channel_signal}")
    decode_transmissions(num_stations, codes, channel_signal,
bit_length)


def decode_transmissions(num_stations, codes, channel_signal,
bit_length):
    """Decode transmissions from channel content for each
station."""
```

```
    for i in range(num_stations):
        received_value = np.dot(codes[i], channel_signal) / 
bit_length
        signal = "1" if received_value > 0 else "0" if 
received_value < 0 else "nothing"
        print(f"Station {i+1} reads: {signal}")


def main():
    num_stations = int(input("Number of stations: "))
    walsh_codes = walsh_code_matrix(num_stations)

    for i in range(num_stations):
        print(f"Station {i+1} code: {walsh_codes[i]}")

    while True:
        simulate_transmission(num_stations, walsh_codes)
        if input("Enter 'x' to exit, or any other key to continue: 
").strip().lower() == 'x':
            break


# Run the main function to initiate the simulation
if __name__ == "__main__":
    main()
```

Output

Number of stations: 4

Station 1 code: [1 1 1 1]

Station 2 code: [1 -1 1 -1]

Station 3 code: [1 1 -1 -1]

Station 4 code: [1 -1 -1 1]

Station 1 is sending: 0

Station 2 is sending: 0

Station 3 is sending: 0

Station 4 is sending: 1

Channel content: [-2. -2. -2. 2.]

Station 1 reads: 0

Station 2 reads: 0

Station 3 reads: 0

Station 4 reads: 1

Enter 'x' to exit, or any other key to continue: f

Station 1 is sending: nothing

Station 2 is sending: nothing

Station 3 is sending: 0

Station 4 is sending: nothing

Channel content: [-1. -1. 1. 1.]

Station 1 reads: nothing

Station 2 reads: nothing

Station 3 reads: 0

Station 4 reads: nothing

Enter 'x' to exit, or any other key to continue:

x

Explanation:

In this simulation, we observe how eight stations communicate simultaneously using Walsh codes, a form of Code Division Multiple Access (CDMA). The process unfolds in several stages, with each station assigned a unique Walsh code to transmit signals. Let's go through each stage in detail.

---

1. Transmission:

Each station independently decides whether to send a signal of `1`, `0`, or remain silent (send "nothing"):

- **Sending `1`**: When a station decides to send a `1`, it transmits its unique Walsh code. This code is a vector of bits that represents the station in the shared communication channel.
- **Sending `0`**: If the station decides to send a `0`, it transmits the negated version of its Walsh code. This effectively inverts the Walsh code values, ensuring that it contributes an opposite signal to the channel compared to sending `1`.
- **Sending Nothing**: If the station opts to send nothing, it contributes a zero vector (`[0, 0, …]`) to the channel, making no impact on the overall signal.

In this specific simulation iteration, each station makes a choice:

- **Stations 2, 6, and 7**: These stations decide to send `1`, so they transmit their unique Walsh codes as positive signals to the channel.
- **Stations 3 and 8**: These stations opt to send `0`, so they transmit the negated versions of their Walsh codes, contributing signals that are the opposite of sending `1`.
- **Stations 1, 4, and 5**: These stations send nothing. Their contributions to the channel are `[0, 0, …]`, leaving the overall channel content unaffected by their Walsh codes.

2. Channel Content Calculation:

After all stations decide on their transmissions, the signals from each active station (those sending `1` or `0`) are combined in the channel. This combination represents the total signal content across all bit positions in the channel:

- **Channel Addition**: Each station's transmission, represented by either its Walsh code or its negated Walsh code, is added up in the channel vector. This addition accumulates contributions from multiple stations at each bit position, creating a composite signal.
- **Resulting Channel Vector**: The final channel content is `[1, -1, 3, -3, -1, -3, 1, 3]`. This vector represents the net result of each station's contribution to the channel across all bit positions of the Walsh code. In this example, positive and negative values at each position reflect the stations that sent `1` or `0`, while contributions from stations that sent nothing leave no impact on the channel content.

3. Decoding by Each Station:

After the channel vector is calculated, each station "reads" the channel to decode the signals:

- **Dot Product Calculation**: Each station calculates the dot product of its Walsh code with the channel content. This operation allows the station to identify the influence of its unique Walsh code in the overall channel content.
- **Interpretation of Dot Product**:
  - If the dot product is positive, the station interprets this as a `1`, indicating that its transmitted signal is detected as `1`.
  - If the dot product is negative, the station reads this as `0`, corresponding to a `0` signal.
  - If the dot product is zero, the station reads "nothing," meaning it detects no signal in the channel related to its unique code.
- **Results for Each Station**:
  - **Stations 2, 6, and 7**: These stations correctly detect `1` as a result, as they had originally sent `1`, and the channel content confirms their positive transmission.
  - **Stations 3 and 8**: These stations detect `0`, which corresponds accurately with their decision to send `0` (the negated Walsh code).
  - **Stations 1, 4, and 5**: These stations detect "nothing," as they opted not to transmit any signal. The dot product of their codes with the channel content results in zero, indicating no contribution to the channel.

Analysis

1. Assumptions and Simplifications:

- **Ideal Channel Conditions**: This simulation assumes a perfect channel with no noise, interference, or fading effects that typically impact real-world CDMA systems. In practical applications, signal transmission is often subject to various forms of interference, including thermal noise, signal fading due to distance, and multipath interference. By removing these complexities, the model focuses solely on the encoding and decoding of signals using Walsh codes.
- **Absence of Flow Control and Error Correction**: The simulation omits error detection and correction protocols, which are standard in real CDMA systems to ensure signal integrity. In practical systems, error correction codes such as Reed-Solomon or convolutional codes are often used alongside Walsh codes to manage transmission errors due to interference.
- **Static Walsh Code Assignment**: Each station is assigned a fixed, unique Walsh code, suitable for a static environment with predefined users. In dynamic networks with variable users, code assignment protocols would be necessary to adapt Walsh codes to the changing number of active stations. Real-world CDMA applications, such as mobile networks, use sophisticated algorithms for dynamic code allocation to handle user variability and optimize spectrum utilization.

2. Channel Content Calculation:

- **Sum of Signal Contributions**: Each station transmits a signal represented by `1`, `0`, or remains silent (`"nothing"`), contributing its Walsh code (or its negated form) or no signal, respectively. The overall channel content is calculated by summing these contributions for each bit position, creating a vector that encapsulates all active station signals.
- **Resource Intensity of Matrix Operations**: As the number of stations increases, the Walsh code matrix and corresponding matrix operations (e.g., dot products for decoding) increase in size and complexity. This simulation's approach is computationally efficient

for small networks, but larger networks would require optimized matrix calculations to manage the increased computational load, particularly if processing in real time.

- **Orthogonal Code Properties and Signal Separation**: The orthogonality of Walsh codes is crucial to the separation of signals from different stations. Each Walsh code is mathematically orthogonal to others, which enables clear, interference-free separation of transmissions. This principle is foundational to CDMA, allowing multiple stations to transmit simultaneously without mutual interference.

3. Signal Decoding by Each Station:

- **Decoding Process**: After the combined channel content is calculated, each station decodes its signal by taking the dot product of its unique Walsh code with the channel content vector. This operation effectively isolates the contribution of the station's signal within the composite channel, allowing for accurate detection of what each station transmitted.
- **Accurate Signal Detection**: The orthogonality of Walsh codes enables each station to accurately decode its own signal. A positive dot product indicates a `1`, a negative result indicates a `0`, and a result of zero indicates "nothing." This accuracy showcases how CDMA allows multi-user access to a shared communication channel with minimal interference between stations.
- **Limitations in Signal Clarity**: While the simulation demonstrates clear signal separation due to ideal conditions, real-world CDMA systems may encounter minor crosstalk or interference due to non-ideal factors like noise and signal overlap. Managing these issues typically requires additional error detection and correction mechanisms in real systems to maintain clarity in signal decoding.

4. Resource Considerations:

- **Output Management**: Continuous real-time output can become resource-intensive, especially as the number of stations grows. For large-scale simulations or practical deployments, managing data output efficiently is essential. Instead of real-time printing, logging the results to a file could optimize performance and reduce processing demands.

- **Real-Time User Interaction**: The simulation's iterative design, which pauses for user input at each iteration, aids in controlled testing and observation of individual rounds. However, continuous automation would be preferable for practical applications or extended simulations, allowing for uninterrupted performance evaluations. This could also be helpful in scenarios where different transmission patterns and code assignments need to be tested over long time frames.

5. Practical Implications and Limitations:

- **Comparison to Real CDMA Systems**: Real CDMA systems introduce several additional complexities, including dynamic code assignment, handling of interference, and error-checking protocols, which are not modeled in this simulation. Real-world systems also face challenges like Doppler shifts and time-varying channel conditions, which require robust designs and algorithms to maintain stable communication.
- **Balancing Accuracy and Complexity**: The simulation provides a simplified but instructive view of CDMA functionality. By focusing on core principles without additional complexities, the model demonstrates CDMA's potential to manage multi-user channels effectively. However, real-world applications would need to address factors such as timing synchronization, noise resilience, and adaptive power control to ensure accuracy in diverse environments.
- **Potential for Extensions and Realistic Modeling**: This project could be expanded to simulate real-world conditions more closely. Future enhancements could include adding noise to the channel, implementing dynamic Walsh code assignment to adapt to changes in active stations, and integrating error correction protocols. These improvements would provide a more comprehensive view of CDMA's capabilities and limitations in practical communication systems.

Conclusion:

This simulation illustrates how Walsh code-based CDMA allows each station to distinguish its transmission within a shared channel, even when multiple stations transmit simultaneously. By encoding each station's signal with a unique Walsh code, the system enables reliable decoding without interference, preserving each station's transmission integrity. The Walsh code properties enable the separation of multiple signals through simple dot product calculations, showcasing CDMA's capacity to handle concurrent transmissions across several stations effectively.

The assignment of implementing CDMA using Walsh codes offers a valuable learning experience in the fundamentals of multi-user communication. Through this simulation, we observe how multiple stations can communicate over a shared channel without interference, thanks to the orthogonal properties of Walsh codes. This demonstrates one of the core strengths of CDMA technology: the ability to achieve simultaneous communication in multi-user environments.