1. **Xử lý tín hiệu là gì. Phân loại. Tại sao phải xử lý tín hiệu**

- Xử lý tín hiệu là quá trình biến đổi, sửa đổi hoặc tách rời tín hiệu để thu được thông tin hữu ích từ tín hiệu đó, liên quan đến việc xử lý và phân tích các tín hiệu được thu thập từ nhiều nguồn khác nhau.

Xử lý tín hiệu thường được thực hiện để cải thiện chất lượng, trích xuất thông tin, giảm nhiễu hoặc thay đổi đặc tính của tín hiệu.

- Phân loại xử lý tín hiệu:

* Xử lý tín hiệu số (Digital Signal Processing - DSP): quá trình xử lý tín hiệu sử dụng các thuật toán số học trên máy tính hoặc thiết bị số hóa; thường áp dụng cho tín hiệu được biểu diễn dưới dạng số học, chẳng hạn như âm thanh, video hoặc dữ liệu mạng.
* Xử lý tín hiệu analog (Analog Signal Processing): quá trình xử lý tín hiệu analog liên quan đến việc thay đổi các thông số của tín hiệu analog, chẳng hạn như tần số, biên độ hoặc pha; thường được thực hiện bằng các mạch điện tử analog, như bộ khuếch đại, bộ điều chỉnh tần số hoặc bộ lọc.
* Ngoài ra còn có
* Xử lý tín hiệu hình ảnh: liên quan đến xử lý và phân tích hình ảnh số; bao gồm các công nghệ như xử lý ảnh số, nhận dạng khuôn mặt, nén hình ảnh và tái tạo hình ảnh.
* Xử lý tín hiệu âm thanh: xử lý và phân tích âm thanh; bao gồm các nhiệm vụ như xử lý tín hiệu âm thanh, nhận dạng giọng nói, nén âm thanh và tái tạo âm thanh.

- Tại sao phải xử lý tín hiệu (XLTH)?

* Xử lý tín hiệu được thực hiện để cải thiện chất lượng và rút trích thông tin từ tín hiệu ban đầu.
* Trong một số trường hợp, tín hiệu ban đầu có thể bị nhiễu, mờ, hoặc chứa thông tin không mong muốn. Bằng cách sử dụng các kỹ thuật xử lý tín hiệu, có thể giảm nhiễu, tăng cường tín hiệu mong muốn và tìm kiếm thông tin quan trọng.
* Xử lý tín hiệu cho phép thay đổi đặc tính của tín hiệu để đáp ứng các yêu cầu cụ thể của ứng dụng.
* Hệ thống truyền thông tin số

Ảnh có chứa biểu đồ, Kế hoạch, Hình chữ nhật, thiết kế

Mô tả được tạo tự động

1. **Có mấy loại điều chế. Tại sao phải điều chế**

- Có hai loại điều chế chính là điều chế tương tự và điều chế số.

* Điều chế tương tự: Điều chế tương tự là quá trình biến đổi tín hiệu thông tin thành dạng sóng mang analog. Trong điều chế analog, thông tin được biểu diễn dưới dạng biến thiên liên tục trong thời gian và biên độ của sóng mang.

Các loại điều chế analog phổ biến bao gồm

* AM (Amplitude Modulation)
* FM (Frequency Modulation)
* PM (Phase Modulation).

Điều chế analog thích hợp cho truyền dẫn âm thanh và video truyền thống.

|  |  |
| --- | --- |
| * Điều chế số: Điều chế số là quá trình biến đổi tín hiệu thông tin thành dạng số hóa trước khi truyền. Trong điều chế số, thông tin được mã hóa thành các bit và truyền qua kênh truyền thông số.   Các loại điều chế số phổ biến bao gồm   * ASK (Amplitude Shift Keying) * FSK (Frequency Shift Keying) * PSK (Phase Shift Keying) * QAM (Quadrature Amplitude Modulation). | Ảnh có chứa văn bản, Phông chữ, hàng, biểu đồ  Mô tả được tạo tự động |

Điều chế số thích hợp cho truyền dữ liệu kỹ thuật số, mạng viễn thông và truyền thông không dây.

- Tại sao phải điều chế:

* Điều chế là quá trình biến đổi tín hiệu thông tin để nó có thể truyền đi thông qua các phương tiện truyền thông như sóng radio, cáp mạng hoặc không gian không dây.
* Điều chế cho phép tín hiệu thông tin được chuyển đổi thành dạng phù hợp để truyền qua các kênh truyền thông và được tái tạo ở điểm đích.
* Điều chế cũng cho phép định dạng và phân biệt giữa các tín hiệu khác nhau, như giọng nói, dữ liệu, hình ảnh, để chúng có thể được truyền và nhận dễ dàng.
* Điều chế giúp tăng hiệu suất truyền thông và giảm nhiễu và suy hao trong quá trình truyền.

1. **OFDM, trực giao là gì, có những tính chất nào,…**

* ***Ghép kênh***: Mang nhiều tín hiệu trên một phương tiện

+ Ghép kênh phân chia theo tần số (FDM)

+ Ghép kênh phân chia theo thời gian (TDM)

+ Ghép kênh phân chia theo mã (CDM)

* ***OFDM*** (Orthogonal Frequency Division Multiplexing) là một dạng đặc biệt của phép điều chế đa sóng mang thông thường FDM với các sóng mang phụ được lựa chọn sao cho mỗi sóng mang phụ là trực giao với các sóng mang phụ còn lại.
* Điều chế đa sóng mang
* OFDM chia thành N luồng dữ liệu song song – gọi là subcarrier. Các subcarrier này được gửi đồng thời và không gây nhiễu cho nhau
* OFDM cho phép chồng chéo làm tăng đáng kể dung lượng.

- ***Trực giao (Orthogonal)*** trong OFDM chỉ ra rằng các subcarrier trong hệ thống OFDM được thiết kế sao cho chúng là trực giao với nhau => các subcarrier không gây nhiễu cho nhau khi truyền qua kênh truyền thông.

* Các subcarrier được thiết kế sao cho phổ của chúng không giao nhau, và tín hiệu trên mỗi subcarrier được đồng bộ hóa một cách chính xác để tránh xung đột và giao thoa giữa các tín hiệu.
* ***Trực giao của 2 tín hiệu***

Về mặt toán học xét tập các tín hiệu *Ψ* với là phần tử thứ p của tập, điều kiện để các tín hiệu trong tập ý trực giao đôi một với nhau là

trong đó là liên hợp phức của . Khoảng từ a đến b là chu kỳ của tín hiệu, k là hằng số

- ***Tín hiệu OFDM***

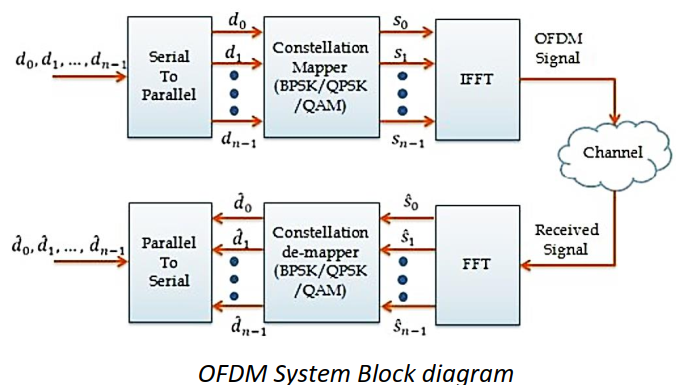
Ảnh có chứa ảnh chụp màn hình, văn bản, biểu đồ, hàng

Mô tả được tạo tự động

*Biểu diễn tần số-thời gian của tín hiệu OFDM*

- Một số tính chất quan trọng của OFDM bao gồm:

* Chống nhiễu tốt: Nhờ tính chất trực giao của các subcarrier, OFDM có khả năng chống lại nhiễu tốt. Khi một subcarrier gặp nhiễu, chỉ một phần nhỏ tín hiệu trên subcarrier đó bị ảnh hưởng, trong khi các subcarrier khác vẫn giữ nguyên tính chất trực giao.
* Tính hiệu quả tần số: OFDM chia tín hiệu truyền thành nhiều subcarrier có tần số thấp, giúp tận dụng tốt băng thông và truyền thông hiệu quả trong môi trường truyền không dây.
* Khả năng xử lý đa đường: OFDM có khả năng xử lý tốt trong môi trường có nhiều đường truyền và tín hiệu phản xạ. Truyền tín hiệu trên nhiều subcarrier, OFDM có khả năng xử lý các hiện tượng như suy hao đa đường và nhiễu tín hiệu phản xạ.
* Dễ dàng triển khai: OFDM có các thuật toán xử lý tương đối đơn giản và có thể triển khai bằng phần cứng hoặc phần mềm.
* Triển khai OFDM: Biến đổi IFFT/FFT



1. **Biến đổi Fourier**

* ***Định nghĩa biến đổi Fourier theo thời gian rời rạc***

+ Phổ của một tín hiệu theo thời gian liên tục x(t) là biến đổi Fourier của x(t), viết tắt là FT, được định nghĩa bằng toán học như sau

X(Ω) =

+ Gọi = x(nT) và đặt (ω) = => biến đổi Fourier theo thời gian rời rạc, viết tắt là DTFT, của tín hiệu rời rạc

* ***Áp dụng biến đổi Fourier theo thời gian rời rạc vào hệ thống tuyến tính bất biến***

+ Biểu thức biến đổi Fourier của đáp ứng xung h(n) tính tại ω = ,

H() =

=> biến đổi Fourier theo thời gian rời rạc, viết tắt là DTFT, của tín hiệu rời rạc

+ Vì thế

+ H(ω) là độ khuếch đại phức của hệ thống tuyến tính bất biến rời rạc. Lúc hệ thống được kích thích bởi một tín hiệu điều hòa với tần số góc ω thì tín hiệu này sẽ được khuếch đại bởi H(ω), độ khuếch đại này thay đổi với tần số góc ω => H(ω) cũng được gọi là **đáp ứng tần số của hệ thống.**

1. **Lập trình C**

* ***Các loại kiểu dữ liệu (từ thấp đến cao), kích thước các loại***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Kiểu dữ liệu*** | ***Kiểu dữ liệu*** | ***Bytes*** | ***Format*** | ***Giá trị min*** | ***Giá trị max*** |
| Kiểu kí tự | char | 1 | %c | -27(-128) | 27 -1 (127) |
| unsigned char | 1 | %c | 0 | 28-1 (255) |
| Kiểu số nguyên | short | 2 | %hi | -215 (-32768) | 215 -1 (32767) |
|  | unsigned short | 2 | %hu | 0 | 216 -1 (65535) |
|  | int | 4 | %i | -231 | 231-1 |
|  | unsigned int | 4 | %u | 0 | 232-1 |
|  | long | 8 | %li | -231 | 231-1 |
|  | unsigned long | 8 | %lu | 0 | 232-1 |
| Kiểu số thực | float | 4 | %f %e %g |  |  |
|  | double | 8 | %lf %le %lg |  |  |
|  | long double | 12 | %Lf %Le  %Lg |  |  |
| Kiểu Boolean và void | • Trong C:  – Không có kiểu boolean  – Kiểu int được sử dụng thay thế cho giá trị boolean. 0 (số  không) biểu diễn False, tất cả các giá trị còn lại được xem  như biểu diễn giá trị True.  • Kiểu void:  – Hàm mà không trả về bất cứ kiểu cụ thể nào  – Kiểu tổng quát sử dụng để ép kiểu | | | | |

* Thường ta hay sử dụng kiểu double.
* Giá trị Max/Min được định nghĩa trong “float.h”
* Bảng mã ASCII

|  |  |  |  |
| --- | --- | --- | --- |
| **Hệ 2** | **Hệ 10** | **Hệ 16** | **Hiển thị ra được** |
| **011 0000** | 48 | 30 | 0 |
| **100 0001** | 65 | 41 | A |
| **110 0001** | 97 | 61 | a |

* ***Toán tử & và &&, | và ||,…***
* & và &&:
* & là toán tử lấy bit "và" (AND) thực hiện phép toán logic AND bit trên các bit tương ứng của hai toán hạng.
* Toán tử && (and) sẽ trả về giá trị:

- true: nếu hai giá trị là true.

- false: nếu hai giá trị là false, hoặc một giá trị là true và một giá trị là false

* Ví dụ:

- 10&14 trả về 10. Đổi giá trị của 10 và 14 từ hệ cơ số 10 sang hệ đếm cơ số 2. Giá trị a = 10 = 1010 còn giá trị của biến b = 14 = 1110, kết quả của toán tử & là: 1010 = 10 => kết quả là 10

- 10 &&14 trả về 1. Do a = 10 và b = 14 đều khác không => kết quả = true (hay có giá trị là 1)

* | và ||
* | là toán tử logic "hoặc" (OR) thực hiện phép toán logic OR bit trên các bit tương ứng của hai toán hạng.
* Toán tử || (or) sẽ trả về giá trị:

- true: nếu hai giá trị là true, hoặc một giá trị là true và một giá trị là false

- false: nếu hai giá trị là false.

* ***Tham chiếu, tham trị***

+ Truyền tham chiếu (pass by reference ) là truyền địa chỉ ô nhớ của biến, do đó khi thay đổi giá trị của biến bên trong phương thức thì giá trị của biến cũng bị thay đổi bên ngoài phương thức.

+ Truyền tham trị (pass by value) là truyền giá trị của biến (không phải là địa chỉ ô nhớ), khi đó phương thức sẽ tự động tạo ra một địa chỉ ô nhớ mới để lưu trữ giá trị này, do đó nó chỉ được thay đổi trong phương thức hiện hành và giá trị của biến không bị thay đổi bên ngoài phương thức hiện hành.

* ***Cấu trúc điều khiển rẽ nhánh***

• Toán tử if

• Toán tử if-else

• Toán tử switch

* ***Vòng lặp:***

• Toán tử for

• Toán tử while

• Toán tử do ... while

* ***Hàm***
* ***Mảng***
* ***Con trỏ***
* ***Struct***
* ***Heap, stack,…***